ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

ΑΓΓΕΛΗΣ ΕΥΑΓΓΕΛΟΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπουσα

Κοζύρη Μαρία

Λαμία, 2021

UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

INFORMATICS AND COMPUTATIONAL BIOMEDICINE

OBJECT DETECTION

ANGELIS EVANGELOS

Master thesis

Koziri Maria

Lamia, 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ ΚΑΤΕΥΘΥΝΣΗ
«ΠΛΗΡΟΦΟΡΙΚΗ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΣΦΑΛΕΙΑ, ΔΙΑΧΕΙΡΙΣΗ ΜΕΓΑΛΟΥ ΟΓΚΟΥ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ»

ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

ΑΓΓΕΛΗΣ ΕΥΑΓΓΕΛΟΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπουσα
Κοζύρη Μαρία

Λαμία, 2021

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο [«τίτλος εργασίας»] αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η ΔΗΛΩΝ/-ΟΥΣΑ

Ημερομηνία

Υπογραφή

ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

ΑΓΓΕΛΗΣ ΕΥΑΓΓΕΛΟΣ

Τριμελής Επιτροπή:

Δρ. Κοζύρη Μαρία, (επιβλέπουσα)	Επίκουρη Καθηγήτρια, Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Πανεπιστήμιο Θεσσαλίας
Δαδαλιάρης Αντώνιος,	Επίκουρος Καθηγητής, Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Πανεπιστήμιο Θεσσαλίας
Τζιρίτας Νικόλαος,	Επίκουρος Καθηγητής, Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Για την εκπόνηση της παρούσας διπλωματικής εργασίας θα ήθελα να εκφράσω την ειλικρινή ευγνωμοσύνη μου στην επιβλέπουσα της παρούσης μεταπτυχιακής εργασίας, επίκουρη καθηγήτρια κυρία Κοζύρη Μαρία για τη συμβολή της σε αυτή την προσπάθεια.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξη και την αγάπη τους όλα αυτά τα χρόνια.

Τέλος, θα ήθελα να ευχαριστήσω τη φίλη μου Μυλωνή Βασιλική για τη πολύτιμη βοήθεια και τη ψυχική συμπαράστασή της.

Περίληψη

Αυτή η διπλωματική εργασία, έχει σκοπό τη συσσώρευση γνώσεων για τη καταλληλότερη εφαρμογή πάνω στην ανίχνευση αντικειμένων. Αρχικά, παρουσιάζουμε κάποιες βασικές έννοιες και υπόβαθρο που χρειάζεται να έχει ο αναγνώστης για τη παρουσίαση των μοντέλων ανίχνευσης αντικειμένων που θα αναφερθούν και ακολουθεί η ανάλυση των μοντέλων ανίχνευσης αντικειμένων. Έπειτα, γίνεται η εκπαίδευση και υλοποίηση εφαρμογής του Faster R-CNN σε δικό μας dataset, ειδικά διαμορφωμένο, για να μπορέσουμε να εξάγουμε δεδομένα για τη ταχύτητά του και την ακρίβειά του με χρήση CPU και GPU. Ακόμη γίνεται ανασκόπηση σε αποτελέσματα εφαρμογών ανίχνευσης αντικειμένων όπου έχουν γίνει σε μεγάλο όγκο δεδομένων για την εγκυρότητα των αποτελεσμάτων που μας παρέχουν. Εν κατακλείδι, γίνεται η παρουσίαση των συμπερασμάτων μας.

Λέξεις Κλειδιά

Ανίχνευση Αντικειμένων, Μηχανική Μάθηση, Συνελικτικά Νευρωνικά Δίκτυα (CNN), Faster R-CNN, You Only Look Once (YOLO),

Abstract

This master thesis aims to accumulate knowledge for the most appropriate application on object detection. First, we present some basic meanings and background that the reader needs to have for the presentation of the object detection models that will be mentioned and follows the analysis of the object detection models. Then, the training and implementation of the Faster R-CNN application is done in our own dataset, specially configured, so that we can export data on its speed and accuracy using CPU and GPU. We are also reviewing the results of object detection applications where a large amount of data has been made on the validity of the results, they provide us. In conclusion, our conclusions are presented.

Keywords

Object Detection, Machine Learning, Convolutional Neural Networks (CNN), Faster R-CNN, You Only Look Once (YOLO),

Πίνακας περιεχομένων

Εισαγωγή

Το Computer Vision ή στα ελληνικά "υπολογιστική όραση" έχει εξαπλωθεί πλέον πάρα πολύ με εφαρμογές στην αναζήτηση, την κατανόηση εικόνων, την χαρτογράφηση, την ιατρική, τα drones αλλά και την αυτοκινητοβιομηχανία. Βασικό κομμάτι όλων αυτών των εφαρμογών είναι το visual recognition, δηλαδή επίλυση προβλημάτων που αφορούν ταυτοποίηση μέσω εικόνας ή video. Για παράδειγμα, αναγνώριση προσώπου (face recognition), η ανίχνευση αντικειμένων (object detection & recognition) και δραστηριοτήτων (activity recognition). Τα τελευταία χρόνια οι εξελίξεις στο πεδίο των νευρωνικών δικτύων (ή αλλιώς deep learning) βελτίωσαν πάρα πολύ την απόδοση των συστημάτων υπολογιστικής όρασης, με αποτέλεσμα να αναπτύσσονται ήδη πολλές εφαρμογές (surveillance, healthcare αλλά και ρομποτικής). Κάποιες είναι αμφιλεγόμενες, όπως για παράδειγμα, η αναγνώριση προσώπου με την οποία μπορεί κανείς να παρακολουθεί και να ταυτοποιεί άτομα, ενώ υπάρχουν κι άλλες πολύ πιο χρήσιμες και πολιτικά ουδέτερες όπως οι αυτοματοποιημένες τεχνικές αναγνώρισης όγκου ή πολύποδα που συμβάλλει στην αντιμετώπιση του καρκίνου.

Στις μέρες μας η ανίχνευση αντικειμένων είναι όλο και πιο δημοφιλής. Μια τυπική τεχνική που χρησιμοποιείται σε μια εικόνα για ανίχνευση αντικειμένου είναι η σάρωσή της από ένα κελί σε όλη την πιθανή έκταση της εικόνας και σε όλα τα πιθανά μεγέθη. Αλλά αυτή η τεχνική έχει μεγάλο μέγεθος επεξεργασίας και πολλές προβλέψεις ότι το στοιχείο ανήκει σε μία κατηγορία και αυτό δεν είναι ορθό (false positives, FP). Επίσης, μια άλλη τεχνική που προϋποθέτει να γνωρίζουμε τα μεγέθη των αντικειμένων που ανιχνεύουμε (μέγεθος, σχήμα ή χρώμα), είναι η τεχνική της μηχανικής μάθησης.

Η δύναμη των αλγορίθμων ανίχνευσης αντικειμένων είναι μεγάλη. Ενώ, οι εφαρμογές ανίχνευσης αντικειμένων καλύπτουν πολλές και διαφορετικές βιομηχανίες, από την 24ωρη παρακολούθηση έως την ανίχνευση οχημάτων σε πραγματικό χρόνο σε έξυπνες πόλεις. Εν ολίγοις, αυτοί είναι ισχυροί αλγόριθμοι βαθιάς μάθησης.

Κεφάλαιο 1

Θεωρητικό υπόβαθρο

Μηχανική όραση

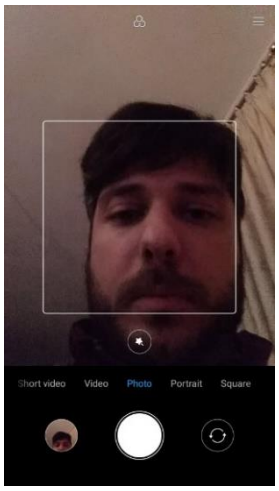
Η μηχανική όραση είναι η επιστήμη και η τεχνολογία μέσω της οποίας οι μηχανές αντιλαμβάνονται το περιβάλλον μέσω πληροφορίας εικόνας. Ως επιστημονική αρχή, η μηχανική όραση ασχολείται με τη θεωρία κατασκευής τεχνητών συστημάτων που λαμβάνουν πληροφορίες από εικόνες ή πολυδιάστατα δεδομένα. Ένα σημαντικό μέρος της τεχνητής νοημοσύνης ασχολείται με το σχεδιασμό ή τη μελέτη ενός συστήματος το οποίο μπορεί να εκτελεί ενέργειες όπως η αυτόνομη μετακίνηση ενός ρομπότ μέσα σε κάποιο περιβάλλον. Αυτός ο τύπος επεξεργασίας τυπικά χρειάζεται δεδομένα εισόδου που παρέχονται από ένα σύστημα μηχανικής όρασης, που λειτουργεί ως οπτικός αισθητήρας και παρέχει πληροφορίες υψηλού επιπέδου για το περιβάλλον και το ρομπότ.

Βαθιά μάθηση

Η βαθιά μάθηση (Deep Learning) είναι μια τεχνική μηχανικής μάθησης που διδάσκει στους υπολογιστές να κάνουν ό,τι έρχεται φυσικά στον άνθρωπο και είναι η πρόοδος των απλών νευρωνικών δικτύων. Για παράδειγμα, η βαθιά μάθηση είναι μια βασική τεχνολογία πίσω από τα αυτοκίνητα χωρίς οδηγό, επιτρέποντάς τους να αναγνωρίσουν μια πινακίδα στάσης ή να διακρίνουν έναν πεζό από έναν λαμπτήρα. Είναι το κλειδί για τον φωνητικό έλεγχο σε καταναλωτικές συσκευές όπως τηλέφωνα, tablet, τηλεοράσεις και ηχεία ανοιχτής ακρόασης.
Στη βαθιά μάθηση, ένα μοντέλο υπολογιστή μαθαίνει να εκτελεί εργασίες ταξινόμησης απευθείας από εικόνες, κείμενο ή ήχο. Τα μοντέλα βαθιάς μάθησης μπορούν να επιτύχουν ακρίβεια τελευταίας τεχνολογίας, μερικές φορές υπερβαίνοντας την απόδοση σε ανθρώπινο επίπεδο. Τα μοντέλα εκπαιδεύονται χρησιμοποιώντας ένα μεγάλο σύνολο δεδομένων με ετικέτες και αρχιτεκτονικές νευρωνικών δικτύων που περιέχουν πολλά επίπεδα. Ο συνδυασμός πολλών επιπέδων των δικτύων με τη λήψη πολυδιάστατων δεδομένων απαιτεί αρκετό χρόνο για να γίνει η επεξεργασία τους. Ο χρόνος αυτός έχει βελτιωθεί σημαντικά λόγω των πολυπύρηνων αρχιτεκτονικών που έχουν δημιουργηθεί και των (GPUs) μονάδων γραφικής επεξεργασίας.

Ανίχνευση αντικείμενων (object detection)

Οι τεχνικές όρασης σε υπολογιστή για τον εντοπισμό αντικειμένων μέσα από εικόνες ή οι τεχνικές ανίχνευσης αντικειμένων από βίντεο, εκπαιδεύουν μοντέλα πρόβλεψης ή χρησιμοποιούν αντιστοίχιση προτύπων για τον εντοπισμό και την ταξινόμηση αντικειμένων ανίχνευσης. Η ανίχνευση αντικειμένων είναι το κλειδί της τεχνολογίας πίσω από εφαρμογές όπως συστήματα ανάκτησης εικόνας, παρακολούθησης βίντεο και προηγμένα συστήματα βοήθειας προγραμμάτων οδήγησης. Υπάρχει μια ποικιλία από τεχνικές που μπορούν να χρησιμοποιηθούν για την εκτέλεση ανίχνευσης αντικειμένων. Οι τεχνικές γενικά εμπίπτουν σε τρεις κύριες κατηγορίες ανίχνευσης αντικειμένων. Αυτές είναι η ανίχνευση αντικειμένων βαθιάς μάθησης (deep learning), η μηχανική μάθηση (machine learning) και η ανίχνευση αντικειμένων χρησιμοποιώντας κλασικές τεχνικές όρασης υπολογιστή. Οι τεχνικές βασισμένες σε βαθιά μάθηση είναι οι πιο δημοφιλείς όπως το R - CNN και YOLO. Σε αυτές θα πρέπει να χρησιμοποιούνται συνελικτικά τα νευρωνικά δίκτυα για να μάθουν τα απαραίτητα χαρακτηριστικά για την ανίχνευση αντικειμένων. Οι προσεγγίσεις μηχανικής μάθησης χρησιμοποιούν εξαγωγή χαρακτηριστικών πριν εκπαιδεύσουν έναν ταξινομητή για να προσδιορίσουν τα αντικείμενα. Τέλος οι πιο παραδοσιακοί μέθοδοι όρασης σε υπολογιστή μπορεί να είναι επαρκείς, ανάλογα με την εφαρμογή.



Εικόνα 1

Το πρώτο βήμα για τη χρήση της βαθιάς μάθησης για την ανίχνευση αντικειμένων είναι να επισημάνουμε δείγματα του τύπου αντικειμένου που θέλουμε να αναγνωρίσουμε, εκπαιδεύοντας ένα προβλεπόμενο μοντέλο για την ανίχνευση αντικειμένων. Συνήθως απαιτεί χιλιάδες ή και εκατομμύρια δείγματα με ετικέτες, οι διαδραστικές εφαρμογές μπορούν να μας βοηθήσουν να αυτοματοποιήσουμε την επισήμανση αντικειμένων και εικόνων ή βίντεο. Αυτό

μας βοηθά να εστιάσουμε περισσότερη προσπάθεια στην ανάπτυξη του αλγορίθμου ανίχνευσης αντικειμένων παρά στην προετοιμασία εκπαίδευσης δεδομένων. Οι τρέχουσες προσεγγίσεις επικεντρώνονται σήμερα στις πληροφορίες από άκρο σε άκρο που έχει βελτιώσει σημαντικά την απόδοση και βοηθάει επίσης στην ανάπτυξη περιπτώσεων χρήσης σε πραγματικό χρόνο. Οι εφαρμογές ανίχνευσης αντικειμένων είναι πιο εύκολο να αναπτυχθούν τα τελευταία χρόνια.

Διεπαφή προγραμματισμού εφαρμογών (API)

Το API σημαίνει διεπαφή προγραμματισμού εφαρμογών. Ένα API παρέχει στους προγραμματιστές ένα σύνολο κοινών λειτουργιών, ώστε να μην χρειάζεται να γράφουν κώδικα από την αρχή. Σκεφτείτε ένα API όπως το μενού σε ένα εστιατόριο που παρέχει μια λίστα με πιάτα μαζί με μια περιγραφή για κάθε πιάτο. Όταν καθορίζουμε ποιο πιάτο θέλουμε, το εστιατόριο λειτουργεί και μας παρέχει έτοιμα πιάτα. Δεν ξέρουμε ακριβώς πώς το εστιατόριο ετοιμάζει αυτό το φαγητό και δεν το χρειαζόμαστε.

Κατά μία έννοια, το API έχει εξαιρετική εξοικονόμηση χρόνου. Προσφέρει επίσης ευκολία στους χρήστες σε πολλές περιπτώσεις. Για παράδειγμα οι χρήστες του Facebook, εκτιμούν τη δυνατότητα σύνδεσης σε πολλές εφαρμογές και ιστότοπους, χρησιμοποιώντας το αναγνωριστικό τους στο Facebook. Αυτό επιτυγχάνεται με το API του Facebook.

Έτσι, θα κάνουμε χρήση του TensorFlow API που αναπτύχθηκε για την ανίχνευση αντικειμένων και εφαρμόστηκε στο 3ο Κεφάλαιο.

Dataset και νευρωνικό δίκτυο

Για βαθιά μάθηση, το σύνολο δεδομένων και το νευρωνικό δίκτυο είναι δύο σημαντικά μέρη. Το σύνολο δεδομένων είναι το καύσιμο για βαθιά μάθηση, έτσι ώστε ο αριθμός και η ποιότητα του συνόλου δεδομένων θα επηρεάσει την ακρίβεια της εξόδου του νευρωνικού δικτύου και την επιλογή του νευρωνικού δικτύου ή η αρχιτεκτονική του δικτύου θα επηρεάσει επίσης την ακρίβεια.

Dataset

Το σύνολο δεδομένων (Dataset) είναι ένα από τα θεμέλια της βαθιάς μάθησης, για πολλούς ερευνητές, ώστε να λάβουν αρκετά δεδομένα για να διεξάγουν το πείραμα μόνοι τους είναι ένα μεγάλο πρόβλημα, οπότε χρειαζόμαστε μια ποικιλία συνόλων δεδομένων

ανοιχτού κώδικα για χρήση από όλους. Κάποια κοινά χρησιμοποιούμενα σύνολα δεδομένων στην όραση του υπολογιστή είναι τα ακόλουθα:

Imagenet

Το σύνολο δεδομένων Imagenet έχει περισσότερες από 14 εκατομμύρια εικόνες καλύπτοντας περισσότερες από 20.000 κατηγορίες. Υπάρχουν περισσότερα από ένα εκατομμύρια φωτογραφίες με σαφείς σχολιασμούς και σχολιασμούς τοποθεσίας αντικειμένων στην εικόνα. Το σύνολο δεδομένων Imagenet, είναι ένα από τα πιο ευρέως χρησιμοποιούμενα σύνολα δεδομένων στον τομέα της βαθιάς μάθησης. Το σύνολο δεδομένων Imagenet είναι λεπτομερές και είναι πολύ εύκολο στη χρήση. Χρησιμοποιείται ευρέως στον τομέα της έρευνας για την όραση υπολογιστών, και έχει γίνει το "τυπικό" σύνολο δεδομένων της τρέχουσας βαθιάς εκμάθησης του τομέα της εικόνας για τον έλεγχο της απόδοσης του αλγορίθμου.

Pascal voc

Το Pascal voc (pattern analysis, statistical modelling and computational learning visual object classes) παρέχει τυποποιημένα σύνολα δεδομένων εικόνας για αναγνώριση ταξινόμησης αντικειμένων και παρέχει ένα κοινό σύνολο εργαλείων για την πρόσβαση συνόλων δεδομένων και σχολιασμών. Το σύνολο δεδομένων Pascal voc περιλαμβάνει 20 κατηγορίες και έχει μια πρόκληση βάσει αυτού του συνόλου δεδομένων. Η πρόκληση του Pascal voc δεν είναι πλέον διαθέσιμη μετά το 2012, αλλά το σύνολο δεδομένων της είναι καλής ποιότητας και καλά χαρακτηρισμένο και επιτρέπει την αξιολόγηση και σύγκριση διαφορετικών μεθόδων. Επειδή το ποσό των δεδομένων του συνόλου δεδομένων Pascal voc είναι μικρό, σε σύγκριση με το σύνολο δεδομένων imagenet, είναι πολύ κατάλληλο για τους ερευνητές για δοκιμή δικτύου προγραμμάτων.

Coco

Το Coco (Common Objects in Context) είναι μια νέα αναγνωριστική εικόνα, τμηματοποίηση και σύνολο δεδομένων υπότιτλων, χορηγούμενο από τη Microsoft. Το σύνολο δεδομένων Coco έχει περισσότερες από 300.000 εικόνες καλύπτοντας 80 κατηγορίες αντικειμένων. Ο ανοιχτός κώδικας αυτού του συνόλου δεδομένων σημειώνει μεγάλη πρόοδο στη σημασιολογική

κατάτμηση τα τελευταία χρόνια, και έχει γίνει ένα "τυπικό" σύνολο δεδομένων για την απόδοση της σημασιολογικής κατανόησης της εικόνας. Επίσης το COCO έχει τη δική του πρόκληση.

Νευρωνικά δίκτυα συνέλιξης

Το νευρωνικά δίκτυα συνέλιξης, επίσης γνωστό ως CNN ή ConvNet, είναι μια ειδική κατηγορία νευρωνικών δικτύων με 3D αρχιτεκτονική που ειδικεύεται στην επεξεργασία δεδομένων που έχουν τοπολογία τύπου πλέγματος, όπως μια εικόνα. Μια ψηφιακή εικόνα είναι μια δυαδική αναπαράσταση οπτικών δεδομένων. Περιέχει μια σειρά pixel διατεταγμένα με τρόπο πλέγματος που περιέχει τιμές pixel για να υποδηλώσει πόσο φωτεινό και ποιο χρώμα πρέπει να είναι κάθε pixel. Σήμερα, τα CNNs έχουν γίνει de facto επιλογή για προβλήματα που αφορούν ταξινόμηση εικόνων μια και έχουν πολύ καλύτερη απόδοση από τις περισσότερες συμβατικές τεχνικές.

Ο ανθρώπινος εγκέφαλος επεξεργάζεται μια τεράστια ποσότητα πληροφοριών. Κάθε νευρώνας λειτουργεί στο δικό του δεκτικό πεδίο και συνδέεται με άλλους νευρώνες με τρόπο που καλύπτει ολόκληρο το οπτικό πεδίο. Ακριβώς όπως κάθε νευρώνας αποκρίνεται σε ερεθίσματα μόνο στην περιορισμένη περιοχή του οπτικού πεδίου που ονομάζεται δεκτικό πεδίο στο βιολογικό σύστημα όρασης, κάθε νευρώνας σε ένα CNN επεξεργάζεται δεδομένα μόνο στο δεκτικό πεδίο του. Τα στρώματα είναι διατεταγμένα με τέτοιο τρόπο ώστε να ανιχνεύουν πρώτα απλούστερα μοτίβα (γραμμές, καμπύλες κ.λπ.) και πιο περίπλοκα μοτίβα (πρόσωπα, αντικείμενα κ.λπ.). Χρησιμοποιώντας ένα CNN, κάποιος μπορεί να επιτρέψει την όραση στους υπολογιστές.

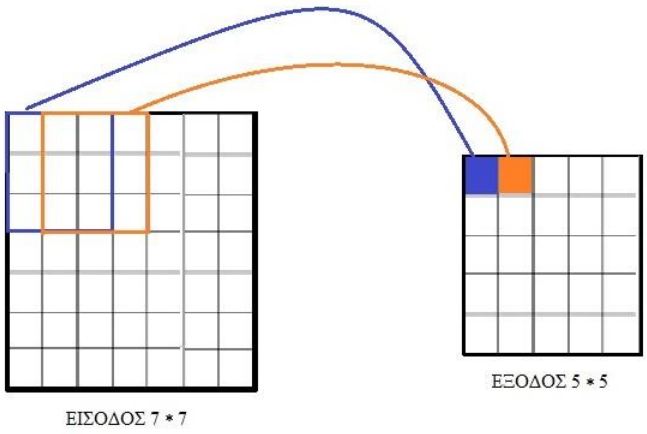
Αρχιτεκτονική του Συνελικτικών Νευρωνικών Δικτύων

Ένα CNN έχει συνήθως τρία επίπεδα: ένα επίπεδο συνέλιξης, ένα επίπεδο συγκέντρωσης και ένα πλήρως συνδεδεμένο επίπεδο.

Επίπεδο συνέλιξης (Convolutional Layer)

Το επίπεδο συνέλιξης είναι το βασικό δομικό στοιχείο του CNN. Μεταφέρει το κύριο μέρος του υπολογιστικού φορτίου του δικτύου. Αυτό το επίπεδο εκτελεί ένα προϊόν κουκκίδων μεταξύ δύο πινάκων, όπου ένας πίνακας είναι το σύνολο των μαθησιακών παραμέτρων που είναι γνωστές ως πυρήνας (φίλτρο) και η άλλη μήτρα είναι το περιορισμένο τμήμα του δεκτικού πεδίου. Ο πυρήνας είναι χωρικά μικρότερος από μια εικόνα, αλλά είναι περισσότερο σε βάθος.

Αυτό σημαίνει ότι, εάν η εικόνα αποτελείται από τρία κανάλια (RGB), το ύψος και το πλάτος του πυρήνα θα είναι χωρικά μικρά, αλλά το βάθος εκτείνεται και στα τρία κανάλια.



Εικόνα 2: Λειτουργία Συνέλιξης

Στην εικόνα 2 τα μπλε και πορτοκαλί πλαίσια που βρίσκονται στην είσοδο του πίνακα $7 * 7$ είναι το φίλτρο $3 * 3$ που από τη μπλε κατάσταση έχει μετακινηθεί ένα βήμα στη πορτοκαλί κατάσταση.

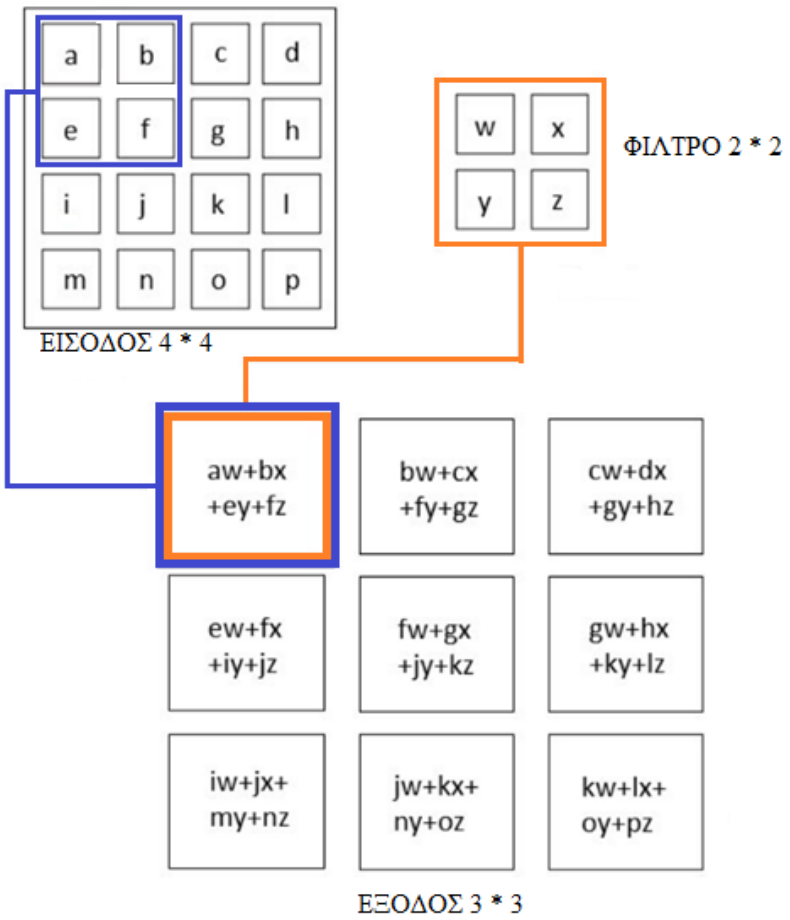
Κατά τη διάρκεια της κίνησης προς τα εμπρός, το φίλτρο (πυρήνας) ολισθαίνει στο ύψος και το πλάτος της εικόνας που παράγει την αναπαράσταση εικόνας αυτής της δεκτικής περιοχής. Αυτό παράγει μια δισδιάστατη αναπαράσταση της εικόνας που είναι γνωστή ως χάρτης ενεργοποίησης που δίνει την απόκριση του πυρήνα σε κάθε χωρική θέση της εικόνας. Το μέγεθος ολίσθησης του πυρήνα ονομάζεται βήμα.

Εάν έχουμε μια είσοδο μεγέθους $W * W * D$ και $Dout$ αριθμός πυρήνων με χωρικό μέγεθος F με διασκελισμό S και ποσότητα padding P, τότε το μέγεθος του όγκου εξόδου μπορεί να προσδιοριστεί με τον ακόλουθο τύπο:

$$W_{out} = \frac{W - F + 2P}{S} + 1 \qquad (1.1)$$

Τύπος για επίπεδο συνέλιξης

Αυτό θα δώσει έναν όγκο εξόδου μεγέθους $W_{out} * W_{out} * Dout$.



Εικόνα 3: Λειτουργία Συνέλιξης

Κίνητρα πίσω από τη συνέλιξη

Η συνέλιξη αξιοποιεί τρεις σημαντικές ιδέες που έδωσαν κίνητρα στους ερευνητές της όρασης υπολογιστών: αραιή αλληλεπίδραση, κοινή χρήση παραμέτρων και ισοδύναμη αναπαράσταση. Ας περιγράψουμε κάθε ένα από αυτά λεπτομερώς.

Τα ασήμαντα επίπεδα νευρωνικών δικτύων χρησιμοποιούν πολλαπλασιασμό μήτρας με έναν πίνακα παραμέτρων που περιγράφουν την αλληλεπίδραση μεταξύ της μονάδας εισόδου και εξόδου. Αυτό σημαίνει ότι κάθε μονάδα εξόδου αλληλεπιδρά με κάθε μονάδα εισόδου. Ωστόσο, τα νευρωνικά δίκτυα συνέλιξης έχουν αραιή αλληλεπίδραση Αυτό επιτυγχάνεται κάνοντας τον πυρήνα μικρότερο από την είσοδο, π.χ., μια εικόνα μπορεί να έχει εκατομμύρια ή χιλιάδες pixel, αλλά κατά την επεξεργασία του χρησιμοποιώντας πυρήνα μπορούμε να εντοπίσουμε σημαντικές πληροφορίες που είναι δεκάδες ή εκατοντάδες pixel. Αυτό σημαίνει ότι πρέπει να αποθηκεύσουμε λιγότερες παραμέτρους που όχι μόνο μειώνουν την απαίτηση μνήμης του μοντέλου αλλά και βελτιώνουν τη στατιστική απόδοση του μοντέλου.

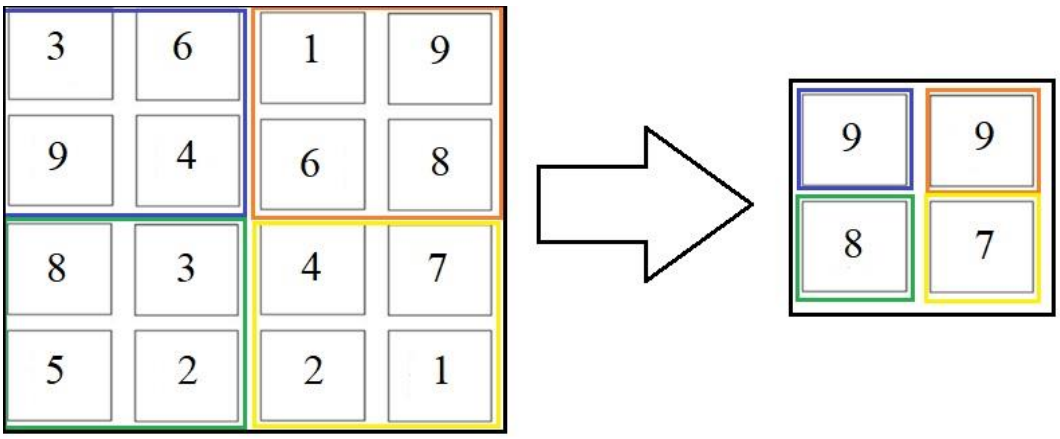
Εάν ο υπολογισμός ενός χαρακτηριστικού σε χωρικό σημείο (x1, y1) είναι χρήσιμος, θα πρέπει επίσης να είναι χρήσιμος σε κάποιο άλλο χωρικό σημείο π.χ. (x2, y2). Αυτό σημαίνει ότι για ένα μονό δισδιάστατο κομμάτι, δηλαδή, για τη δημιουργία ενός χάρτη ενεργοποίησης, οι νευρώνες υποχρεούνται να χρησιμοποιούν το ίδιο σύνολο βαρών. Σε ένα παραδοσιακό νευρωνικό δίκτυο, κάθε στοιχείο της μήτρας βάρους χρησιμοποιείται μία φορά και στη συνέχεια δεν επανεξετάζεται ποτέ, ενώ το δίκτυο συνέλιξης έχει κοινές παραμέτρους, δηλαδή, για τη λήψη εξόδου, τα βάρη που εφαρμόζονται σε μία είσοδο είναι τα ίδια με το βάρος που εφαρμόζεται αλλού.

Λόγω της κοινής χρήσης παραμέτρων, τα επίπεδα του νευρικού δικτύου συνέλιξης θα έχουν μια ιδιότητα ισοδυναμίας με τη μετάφραση. Λέει ότι αν αλλάξαμε την είσοδο κατά κάποιο τρόπο, η έξοδος θα αλλάξει επίσης με τον ίδιο τρόπο.

Επίπεδο συγκέντρωσης (Pooling Layer)

Το επίπεδο συγκέντρωσης αντικαθιστά την έξοδο του δικτύου σε συγκεκριμένες τοποθεσίες, αντλώντας μια συνοπτική στατιστική των κοντινών εξόδων. Αυτό βοηθά στη μείωση του χωρικού μεγέθους της αναπαράστασης, η οποία μειώνει την απαιτούμενη ποσότητα υπολογισμού και βαρών. Η συγκέντρωση επεξεργάζεται σε κάθε κομμάτι της αναπαράστασης ξεχωριστά.

Υπάρχουν πολλές λειτουργίες συγκέντρωσης όπως ο μέσος όρος της ορθογώνιας γειτονιάς, ο κανόνας L2 της ορθογώνιας γειτονιάς και ένας σταθμισμένος μέσος όρος με βάση την απόσταση από το κεντρικό εικονοστοιχείο. Ωστόσο, η πιο δημοφιλής διαδικασία είναι η μέγιστη συγκέντρωση, η οποία αναφέρει τη μέγιστη απόδοση από τη γειτονιά.



Εικόνα 4: Λειτουργία συγκέντρωσης

Η εικόνα 4 αναπαριστά τη μέγιστη λειτουργία συγκέντρωσης. Από το κάθε χρωματικό πλαίσιο του πίνακα εισόδου $4 * 4$ συγκεντρώνεται η μέγιστη τιμή στο πίνακα εξόδου $2 * 2$.

Εάν έχουμε έναν χάρτη ενεργοποίησης μεγέθους $W * W * D$, έναν πυρήνα συγκέντρωσης χωρικού μεγέθους F και το βήμα S, τότε το μέγεθος του όγκου εξόδου μπορεί να προσδιοριστεί με τον ακόλουθο τύπο:

$$W_{out} = \frac{W - F}{S} + 1 \qquad (1.2)$$

Τύπος για padding επίπεδο

Αυτό θα δώσει έναν όγκο εξόδου μεγέθους $W_{out} * W_{out} * D$.

Σε όλες τις περιπτώσεις, η ομαδοποίηση παρέχει κάποια μεταβλητή μεταφράσεων που σημαίνει ότι ένα αντικείμενο θα ήταν αναγνωρίσιμο ανεξάρτητα από το πού εμφανίζεται στο πλαίσιο.

Πλήρως συνδεδεμένο επίπεδο (Fully-Connected Layer)

Οι νευρώνες σε αυτό το επίπεδο έχουν πλήρη συνδεσιμότητα με όλους τους νευρώνες στο προηγούμενο και το επόμενο επίπεδο όπως φαίνεται στο κανονικό FCNN. Αυτός είναι ο λόγος για τον οποίο μπορεί να υπολογιστεί ως συνήθως με πολλαπλασιασμό μήτρας που ακολουθείται από ένα μεροληπτικό αποτέλεσμα.

Το επίπεδο FC βοηθά στη χαρτογράφηση της αναπαράστασης μεταξύ της εισόδου και της εξόδου.

Επίπεδα μη γραμμικότητας

Δεδομένου ότι η συνέλιξη είναι μια γραμμική λειτουργία και οι εικόνες απέχουν πολύ από τη γραμμική, τα στρώματα μη γραμμικότητας τοποθετούνται συχνά αμέσως μετά το συνελικτικό στρώμα για να εισαγάγουν τη μη γραμμικότητα στον χάρτη ενεργοποίησης.

Υπάρχουν διάφοροι τύποι μη γραμμικών λειτουργιών / συναρτήσεις ενεργοποίησης, οι δημοφιλείς είναι:

- **Σιγμοειδής (Sigmoid)**

Η σιγμοειδής μη γραμμικότητα έχει τη μαθηματική μορφή:

$$f(k) = \frac{1}{1 + e^{-k}} \qquad (1.3)$$

Παίρνει έναν πραγματικό αριθμό σε εύρος μεταξύ 0 και 1.

Ωστόσο, μια πολύ ανεπιθύμητη ιδιότητα του σιγμοειδούς είναι ότι όταν η ενεργοποίηση είναι σε κάθε ουρά, η κλίση γίνεται σχεδόν μηδενική. Εάν η τοπική κλίση γίνει πολύ μικρή, τότε στην οπίσθια αναπαραγωγή θα εξαλείψει αποτελεσματικά την κλίση. Επίσης, εάν τα δεδομένα που εισέρχονται στον νευρώνα είναι πάντα θετικά, τότε η έξοδος του σιγμοειδούς θα είναι είτε όλα τα θετικά είτε όλα τα αρνητικά, με αποτέλεσμα μια δυναμική με απότομες εναλλαγές των ενημερώσεων διαβάθμισης για το βάρος.

- **Softmax**

Η συνάρτηση softmax είναι μία γενίκευση της προαναφερόμενης συνάρτησης (σιγμοειδής). Το πλεονέκτημα που έχει είναι η ταξινόμηση πάνω από δύο κατηγοριών. Επομένως δε περιορίζεται σε δυαδική ταξινόμηση. Η μαθηματική μορφή του είναι η εξής:

$$f(k) = \frac{e^{x_j}}{\sum_{j=1}^{k} e^{x_j}} \qquad (1.4)$$

- **Tanh**

Ο Tanh καταγράφει έναν πραγματικό αριθμό στο εύρος [-1, 1]. Όπως το σιγμοειδές, η ενεργοποίηση διαποτίζει, αλλά σε αντίθεση με τους σιγμοειδείς νευρώνες, η έξοδος του είναι μηδενική.

- **ReLU**

Η Ανορθωμένη γραμμική μονάδα (Rectified Linear Unit, ReLU) έχει γίνει πολύ δημοφιλής τα τελευταία χρόνια. Υπολογίζει τη συνάρτηση:

$$f(k) = \max(0, k) \qquad (1.5)$$

Με άλλα λόγια, η ενεργοποίηση είναι απλώς κατώφλι στο μηδέν. Σε σύγκριση με το σιγμοειδές και το tanh, το ReLU είναι πιο αξιόπιστο και επιταχύνει τη σύγκλιση κατά έξι φορές.

Δυστυχώς, ένα μειονέκτημα είναι ότι το ReLU μπορεί να είναι εύθραυστο κατά τη διάρκεια της προπόνησης. Μια μεγάλη κλίση που διατρέχει μπορεί να την ενημερώσει με τέτοιο τρόπο ώστε ο νευρώνας να μην ενημερωθεί ποτέ περαιτέρω. Ωστόσο, μπορούμε να εργαστούμε με αυτόν τον καθορισμό ενός κατάλληλου ποσοστού μάθησης.

Batch Normalization

Η Κανονικοποίηση Παρτίδας (Batch Normalization) είναι μια τεχνική για την εκπαίδευση πολύ βαθιών νευρωνικών δικτύων που τυποποιεί τις εισόδους σε ένα επίπεδο για κάθε μίνι παρτίδα. Η κανονικοποίηση περιορίζει τα χαρακτηριστικά των δεδομένων σε ορισμένες κλίμακες Παλιότερα γινόταν μια προεπεξεργασία πριν αρχίσει η εκπαίδευση, ενώ τώρα μέσω της κανονικοποίησης παρτίδας γίνεται και στα ενδιάμεσα στάδια της εκπαίδευσης. Γίνεται κανονικοποίηση ανά παρτίδα μετά από τα συνελικτικά επίπεδα και πριν τα επίπεδα ενεργοποίησης. Δηλαδή αφαιρεί τη μέση τιμή των χαρακτηριστικών και διαιρεί με την τυπική απόκλιση. Αυτό έχει ως αποτέλεσμα τη σταθεροποίηση της μαθησιακής διαδικασίας, τη καταπολέμηση της υπερπροσαρμογής και τη δραματική μείωση του αριθμού των εποχών εκπαίδευσης που απαιτούνται για την εκπαίδευση βαθιών δικτύων.

Δημοφιλή συνελικτικά νευρωνικά δίκτυα

Μερικά γνωστά συνελικτικά νευρωνικά δίκτυα όπου επιλέγονται βάσει τον τύπο του εκάστοτε προβλήματος που θέλουμε να επιλύσουμε, είναι το ALexNet, το ZF Net, το Inception (V1, V2, V3, V4), το VGG Net και το ResNet.

- **AlexNet:** Αυτό το δίκτυο αναπτύχθηκε από τον Alex Krizhevsky για τον διαγωνισμό ImageNet το 2012, όπου κατέκτησε τη πρώτη θέση, ήταν πρωτοποριακό και δημιούργησε μια τάση στα Συνελικτικά Δίκτυα της Όρασης Υπολογιστών. Επίσης διαθέτει 60 εκατομμύρια παραμέτρους.

- **ZF Net:** Το δίκτυο αυτό αναπτύχθηκε από τους Matthew Zeiler και Rob Fergus για τον διαγωνισμό LSVRC του ImageNet το 2013 και έλαβε την πρώτη θέση. Το ZF Net είναι μια βελτιωμένη εκδοχή του AlexNet, αλλάζοντας κάποιες υπερπαραμέτρους της αρχιτεκτονικής. Εκτενέστερα, επεκτείνανε το μέγεθος του μεσαίου συνελικτικού επιπέδου και ρύθμισαν το άλμα (stride) και το μέγεθος του φίλτρου του πρώτου επιπέδου να είναι μικρότερα.

- **GoogleNet:** Ο Szegedy πήρε τη πρώτη θέση στον διαγωνισμού ILSVRC 2014 με το συγκεκριμένο δίκτυο το οποίο αναπτύχθηκε από την Google. Το μεγάλο προβάδισμα αυτού του δικτύου είναι η ανάπτυξη του ¨Inception Module¨ που μείωσε σημαντικά τις παραμέτρους του στα 4 εκατομμύρια.

- **VGGNet:** Το συγκεκριμένο δίκτυο, αναπτύχθηκε από τους Karen Simonyan και Andrew Zisserman για τον διαγωνισμό LSVRC του ImageNet το 2014 και πήρε την δεύτερη θέση. Ήταν αντίστοιχα αποτελεσματικό με το GoogleNet. Η βασική συνεισφορά του ήταν στο γεγονός ότι το βάθος ενός δικτύου είναι ένα σημαντικό συστατικό για την καλή απόδοση. Η τελική έκδοση αποτελείται από 16 Συνελικτικά/Πλήρη επίπεδα, όπου αποτελείται από ομογενή αρχιτεκτονική και επεξεργάζεται μόνο 3x3 συνελίξεις και 2x2 συγκεντρώσεις. Επειδή έχει πολλές παραμέτρους (138 εκατομμύρια) είναι πολύ απαιτητικό στην μνήμη. Λόγω ότι η πληθώρα των παραμέτρων είναι στο πρώτο πλήρες συνδεδεμένο επίπεδο, έχει αποδειχτεί ότι η αφαίρεση τέτοιων επιπέδων δεν υποβαθμίζει την απόδοση του δικτύου, όπου επιτυγχάνει την μεγάλη μείωση του πλήθους των παραμέτρων.

- **ResNet:** Ο Kaiming He ήταν ο δημιουργός του δικτύου το οποίο κατέκτησε τη πρώτη θέση στον διαγωνισμό ILSVRC 2015. Πρωτοεμφάνισε τις παραλειπούμενες συνδέσεις (skip connections) και εκτεταμένη χρήση κανονικοποίησης. Από την αρχιτεκτονική του δικτύου αυτού λείπουν τα πλήρως συνδεδεμένα επίπεδα και επιτρέπει την ανάπτυξη πολύ βαθύτερων δικτύων, με εκατοντάδες επίπεδα σε σχέση με τα μέχρι πρότινος δίκτυα που είχαν κάποιες δεκάδες επίπεδα. Επίσης, δεν περιέχει πλήρως συνδεδεμένο επίπεδο στο τέλος του δικτύου.

Καταπολέμηση του Overfiting

Overfiting

Όταν το δίκτυο έχει μάθει πάρα πολλά ή πάρα πολλά δεδομένα εκπαίδευσης μαζί με τον θόρυβο οδηγεί σε κακή απόδοση στο δοκιμαστικό σύνολο δεδομένων. Όταν συμβαίνει αυτό, το δίκτυο αποτυγχάνει να γενικεύσει τις δυνατότητες / μοτίβο που βρίσκονται στα δεδομένα εκπαίδευσης. Η υπερφόρτωση (Overfiting) κατά τη διάρκεια της προπόνησης μπορεί να εντοπιστεί όταν το σφάλμα στα δεδομένα εκπαίδευσης μειώνεται σε πολύ μικρή τιμή, αλλά το σφάλμα στα νέα δεδομένα ή τα δεδομένα δοκιμής αυξάνεται σε μεγάλη τιμή.

Underfitting

Το underfitting συμβαίνει όταν το δίκτυο δεν μπορεί να μοντελοποιήσει τα δεδομένα εκπαίδευσης ή δοκιμής που οδηγούν σε συνολική κακή απόδοση. Ο λόγος για το ανεπαρκές

μοντέλο μπορεί να οφείλεται στην περιορισμένη χωρητικότητα του δικτύου, σε περιορισμένο αριθμό χαρακτηριστικών που παρέχονται ως είσοδος στο δίκτυο, θορυβώδη δεδομένα κ.λπ.

Υπερπαράμετροι

Στην μηχανική εκμάθηση των νευρωνικών δικτύων, υπάρχουν κάποιοι παράμετροι. Αυτοί είναι οι συντελεστές του μοντέλου και επιλέγονται από το ίδιο το μοντέλο. Αυτό σημαίνει ότι ο αλγόριθμος, ενώ μαθαίνει, βελτιστοποιεί αυτούς τους συντελεστές (σύμφωνα με μια δεδομένη στρατηγική βελτιστοποίησης) και επιστρέφει μια σειρά παραμέτρων που ελαχιστοποιούν το σφάλμα. Εκτός από τις παραμέτρους έχουμε και κάποιες άλλες παραμέτρους που βοηθούν να είναι αποδοτικότερο το μοντέλο. Αυτές είναι οι υπερπαραμέτροι που τις ρυθμίζουμε εμείς έχοντας στόχο το βέλτιστο αποτέλεσμα του μοντέλου και την αποφυγή του Overfit ή Underfit. Κάποιες από αυτές τις υπερπαραμέτρους είναι οι ακόλουθες:

Learning Rate

Ο ρυθμός εκμάθησης έχει την ιδιότητα να αλλάζει τις παραμέτρους του μοντέλου ανάλογα με το ρυθμό που εκπαιδεύεται το μοντέλο. Όσο πιο μικρός ο ρυθμός εκμάθησης, η εκπαίδευση είναι πιο αργή και με αυτό το τρόπο πετυχαίνει ελάχιστο σφάλμα. Όσο πιο μεγάλος ο ρυθμός εκμάθησης, υπάρχουν μεγάλες αλλαγές στα βάρη, όπου το αποτέλεσμα είναι ότι το δίκτυο μεταπηδά κλάσεις – κατηγορίες, όποτε διορθώνει τα βάρη χωρίς να επιτυγχάνεται η σύγκλιση. Ο αλγόριθμος Gradient Descent έχει στόχο την αλλαγή των βαρών του δικτύου αναλόγως τον ρυθμό εκμάθησης που έχει το δίκτυο.

Momentum

Ο αλγόριθμος Gradient Descent είναι ένας αλγόριθμος βελτιστοποίησης που λειτουργεί βρίσκοντας την κατεύθυνση της πιο απότομης κλίσης στην τρέχουσα κατάστασή του και ενημερώνει την κατάστασή του μετακινώντας προς αυτήν την κατεύθυνση. Ως αποτέλεσμα, σε κάθε βήμα είναι εγγυημένο ότι η τιμή της προς ελαχιστοποίηση συνάρτησης μειώνεται κατά κάθε βήμα. Το πρόβλημα είναι ότι αυτή η κατεύθυνση μπορεί να αλλάξει σημαντικά σε ορισμένα σημεία της λειτουργίας, ενώ η καλύτερη διαδρομή που πρέπει να ακολουθήσουμε συνήθως δεν περιέχει πολλές εναλλαγές. Επομένως, είναι επιθυμητό να κάνουμε τον αλγόριθμο να διατηρεί την κατεύθυνση που έχει ήδη ακολουθήσει για λίγο πριν αλλάξει την κατεύθυνση. Για να γίνει

αυτό, εισάγεται το Momentum. Επίσης, επηρεάζει την τιμή του ρυθμού εκμάθησης με τον ίδιο τρόπο που επηρεάζει ο ρυθμός μάθησης τα βάρη

Early Stopping

Ενώ εκπαιδεύετε ένα νευρωνικό δίκτυο χρησιμοποιώντας έναν αλγόριθμο βελτιστοποίησης όπως το Gradient Descent, οι παράμετροι του μοντέλου (βάρη) ενημερώνονται για τη μείωση του σφάλματος εκπαίδευσης. Στο τέλος κάθε προωθητικής μετάδοσης, οι παράμετροι δικτύου ενημερώνονται για τη μείωση του σφάλματος στην επόμενη επανάληψη. Η υπερβολική προπόνηση μπορεί να έχει ως αποτέλεσμα την υπερβολική προσαρμογή των δεδομένων εκπαίδευσης στο δίκτυο. Η πρόωρη διακοπή (Early Stopping) συσχετίζεται με τον αριθμό των επαναλήψεων που μπορούν να εκτελεστούν πριν το δίκτυο αρχίσει να υπερφορτώνεται. Πόσες επαναλήψεις χρειάζονται ώστε το μοντέλο να είναι το βέλτιστο, δε το γνωρίζουμε. Μόνο εμπειρικά μπορούμε να καταλάβουμε και να επιλέξουμε το ποσό των επαναλήψεων που πρέπει να γίνουν.

Dropout

Το Dropout είναι μια δημοφιλής τεχνική κατά του Overfiting. Αποτρέπει την υπερβολική τοποθέτηση των νευρώνων και παρέχει έναν αποτελεσματικό τρόπο περίπου συνδυασμού εκθετικά πολλών διαφορετικών αρχιτεκτονικών νευρωνικών δικτύων. Ο όρος "Dropout" αναφέρεται σε προσωρινή κατάργηση νευρώνων (κρυφών και ορατών) σε ένα νευρωνικό δίκτυο, μαζί με όλες τις εισερχόμενες και εξερχόμενες συνδέσεις του. Έτσι, οι κόμβοι (νευρώνες) που καταργήθηκαν δε παίρνουν μέρος στην εκπαίδευση της τρέχουσας εποχής και δεν υπάρχει αλλοίωση των τιμών τους. Με το Dropout πετυχαίνουμε καλύτερη ακρίβεια στο test set λόγω του ότι καταργώντας έναν αριθμό κόμβων (νευρώνων), οι εναπομείναντες αναγκάζονται να εκπαιδευτούν λόγω έλλειψη αυτών στο δίκτυο.

Η επιλογή των μονάδων που θα πέσει είναι τυχαία. Στην απλούστερη περίπτωση, κάθε μονάδα διατηρείται με μια σταθερή πιθανότητα p ανεξάρτητη από άλλες μονάδες, όπου το p μπορεί να οριστεί στο 0.5, το οποίο φαίνεται να είναι σχεδόν βέλτιστο για ένα ευρύ φάσμα δίκτυα και εργασίες. Για τις μονάδες εισόδου, ωστόσο, η βέλτιστη πιθανότητα συγκράτησης είναι συνήθως πιο κοντά στο 1 παρά στο 0,5.

Regularization

Αυτή είναι μια μορφή παλινδρόμησης, που περιορίζει / κανονικοποιεί ή συρρικνώνει τις εκτιμήσεις του συντελεστή στο μηδέν. Με άλλα λόγια, αυτή η τεχνική αποθαρρύνει την εκμάθηση ενός πιο περίπλοκου ή ευέλικτου μοντέλου, ώστε να αποφευχθεί ο κίνδυνος υπερβολικής τοποθέτησης (Overfiting) μικραίνοντας τα βάρη. Η κανονικοποίηση, μειώνει σημαντικά τη διακύμανση του μοντέλου, χωρίς ουσιαστική αύξηση της προκατάληψής του. Μαθηματικά μπορεί να συμβολιστεί με το γράμμα λ. Έτσι, η παράμετρος συντονισμού λ, που χρησιμοποιείται στις τεχνικές κανονικοποίησης που περιγράφονται παραπάνω, ελέγχει την επίδραση στην προκατάληψη και τη διακύμανση. Καθώς η τιμή του λ αυξάνεται, μειώνει την τιμή των συντελεστών και μειώνοντας έτσι τη διακύμανση. Μέχρι ένα σημείο, αυτή η αύξηση του λ είναι ωφέλιμη, καθώς μειώνει μόνο τη διακύμανση (αποφεύγοντας έτσι το Overfiting), χωρίς να χάσετε σημαντικές ιδιότητες στα δεδομένα. Αλλά μετά από μια συγκεκριμένη τιμή, το μοντέλο αρχίζει να χάνει σημαντικές ιδιότητες, προκαλώντας προκατάληψη στο μοντέλο και επομένως Underfitting. Επομένως, η τιμή του λ πρέπει να επιλεγεί προσεκτικά. Έτσι, με αυτό τον τρόπο αποφεύγουμε τα μεγάλα βάρη που μας οδηγούν σε λάθος προβλέψεις και τείνουμε προς τα μικρά βάρη όπου βοηθούν το δίκτυο να έχει μια καλύτερη γενική εικόνα στη πρόβλεψη.

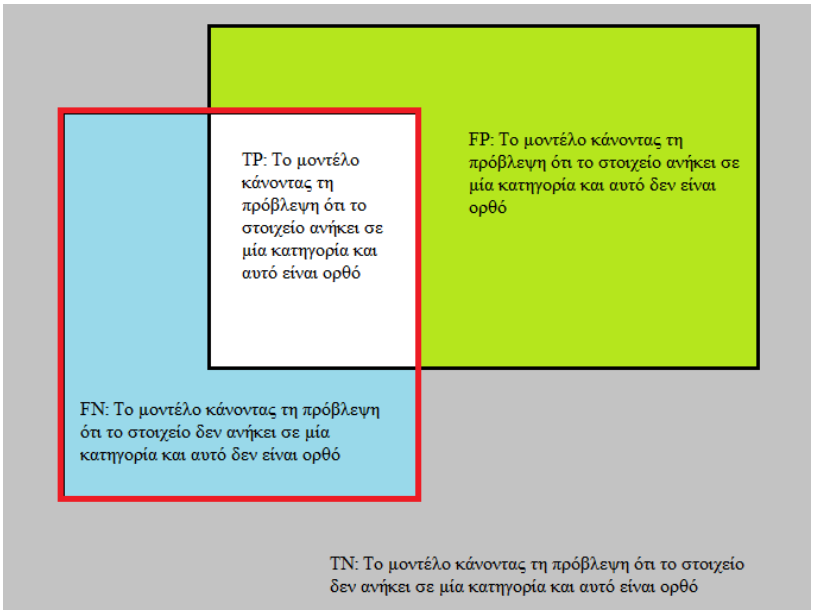
Μετρικές συναρτήσεις

Οι συναρτήσεις αυτές χρησιμοποιούνται για να αξιολογήσουμε κατά πόσο αποδοτικό είναι το μοντέλο μας. Το μοντέλο έχοντας εκπαιδευτεί σε επιβλεπόμενη μάθηση, κάνει τις δικές του προβλέψεις και με τις μετρικές συναρτήσεις θα αξιολογήσουμε τις προβλέψεις του κατά πόσο συμπίπτουν με τις πραγματικές τιμές.

Σε αυτό το σημείο θα αναφέρουμε τέσσερις βασικές έννοιες που αξιοποιούνται στις μετρικές συναρτήσεις:

- True Positive (TP): Το μοντέλο κάνοντας τη πρόβλεψη ότι το στοιχείο ανήκει σε μία κατηγορία και αυτό είναι ορθό.
- True Negative (TN): Το μοντέλο κάνοντας τη πρόβλεψη ότι το στοιχείο δεν ανήκει σε μία κατηγορία και αυτό είναι ορθό.
- False Positive (FP): Το μοντέλο κάνοντας τη πρόβλεψη ότι το στοιχείο ανήκει σε μία κατηγορία και αυτό δεν είναι ορθό.

- False Negative (FN): Το μοντέλο κάνοντας τη πρόβλεψη ότι το στοιχείο δεν ανήκει σε μία κατηγορία και αυτό δεν είναι ορθό.



Εικόνα 5

Στη παραπάνω εικόνα βλέπουμε τις βασικές έννοιες TP, TN, FP, FN. Το κόκκινο πλαίσιο αναπαριστά τις πραγματικές τιμές των δεδομένων, ενώ το μαύρο πλαίσιο αναπαριστά την πρόβλεψη του μοντέλου.

Οι πιο δημοφιλής μετρικές συναρτήσεις απόδοσης είναι οι εξής:

- **Accuracy**

Αυτή η συνάρτηση αξιολογεί στατιστικά κατά πόσο ένα μοντέλο ανιχνεύει ορθά ή καθόλου μια κατάσταση. Συγκεκριμένα, η ακρίβεια είναι η αναλογία των ορθών αποτελεσμάτων (TP και TN) μεταξύ όλων των προβλέψεων ορθών και μη ορθών. Στην ανίχνευση αντικειμένων δε χρησιμοποιείται ιδιαίτερα γιατί έχουμε πληθώρα τιμών TN. Ο τύπος της συνάρτησης είναι ο ακόλουθος:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (1.6)$$

- **Precision**

Η μετρική συνάρτηση της ακρίβειας (precision) χρησιμοποιείται πιο συχνά για την ανίχνευση αντικειμένων από την προηγούμενη συνάρτηση καθώς προσμετρά αποκλειστικά τα θετικά δείγματα αλλά σαν μοναδικό μέτρο σύγκρισης δεν επαρκεί. Συγκεκριμένα, είναι οι ποσοστιαίες επιτυχημένες προβλέψεις μίας κατηγορίας του μοντέλου ως προς το σύνολο των προβλέψεων. Ο τύπος της συνάρτησης είναι ο ακόλουθος:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (1.7)$$

- **Recall**

Η ανάκληση (recall) είναι οι ποσοστιαίες επιτυχημένες προβλέψεις μίας κατηγορίας του μοντέλου ως προς το σύνολο των πραγματικών τιμών. Η μετρική συνάρτηση precision, σαν μοναδικό μέτρο σύγκρισης δεν επαρκεί. Ο τύπος αυτής της συνάρτησης είναι ο ακόλουθος:

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (1.8)$$

- **Τομή προς Ένωση (Intersection over Union)**

Μια πιο ευαίσθητη μετρική συνάρτηση είναι η τομή προς ένωση (Intersection over Union, IoU). Η IoU απεικονίζει τις ποσοστιαίες επιτυχημένες προβλέψεις ως προς το σύνολο των πραγματικών τιμών των δεδομένων με την πρόβλεψη του μοντέλου. Ο τύπος της συνάρτησης είναι ο ακόλουθος:

$$\text{IoU} = \frac{\text{TP}}{\text{TP+FP+FN}} \qquad (1.9)$$

ή αλλιώς

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \qquad (1.10)$$

- **Average Precision**

Η average precision (AP) είναι η πιο αξιοποιήσιμη για ανίχνευση αντικειμένων. Ουσιαστικά για κάθε πρόβλεψη του μοντέλου υπολογίζονται precision και recall και

κατηγοριοποιούνται με αύξουσα ανάκληση. Η AP υπολογίζει την τιμή της precision (p) για κάθε τιμή της ανάκλησης (r) από 0 έως 1 και εκφράζεται ως εξής:

$$\mathrm{AP} = \int_0^1 \mathrm{p(r)dr} \qquad (1.11)$$

Για να θεωρηθεί μια πρόβλεψη σωστή και να συμβάλλει στην AP θα πρέπει να επιτυγχάνει IoU μεγαλύτερη από 0.5. Στην πράξη το ολοκλήρωμα ισοδυναμεί με το έξης άθροισμα της συνάρτησης:

$$\mathrm{AP} = \sum_{\mathrm{k=1}}^{\mathrm{N}} \mathrm{P(k)\Delta r(k)} \qquad (1.12)$$

Όπου N ο αριθμός των προβλέψεων και k οι προβλέψεις μέχρι μια δεδομένη ανάκληση. Σε προβλήματα με πολλαπλές κλάσεις υπολογίζεται για κάθε κλάση η average precision και μετά η mean average precision (mAP) ως μέσος όρος των ξεχωριστών AP.

- **Μέσος όρος της μέσης τιμής της ακρίβειας (mAP)**

Ο μέσος όρος της μέσης τιμής της ακρίβειας (mAP) ή μερικές φορές απλά αναφέρεται ως η μέση ακρίβεια (Average Precision, AP) είναι μια δημοφιλής μέτρηση που χρησιμοποιείται για τη μέτρηση της απόδοσης των μοντέλων που κάνουν εργασίες παραλαβής εγγράφων ή πληροφοριών και εντοπισμού αντικειμένων.

Ο μέσος όρος της μέσης τιμής της ακρίβειας (mAP) ενός συνόλου ερωτημάτων ορίζεται ως εξής:

$$\mathrm{mAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{\mathrm{Q}} \qquad (1.13)$$

Τύπος του μέσου όρου της μέσης τιμής της ακριβείας

όπου Q είναι ο αριθμός των ερωτημάτων στο σύνολο και $\mathrm{AveP(q)}$ είναι η μέση ακρίβεια (AP) για ένα δεδομένο ερώτημα q.

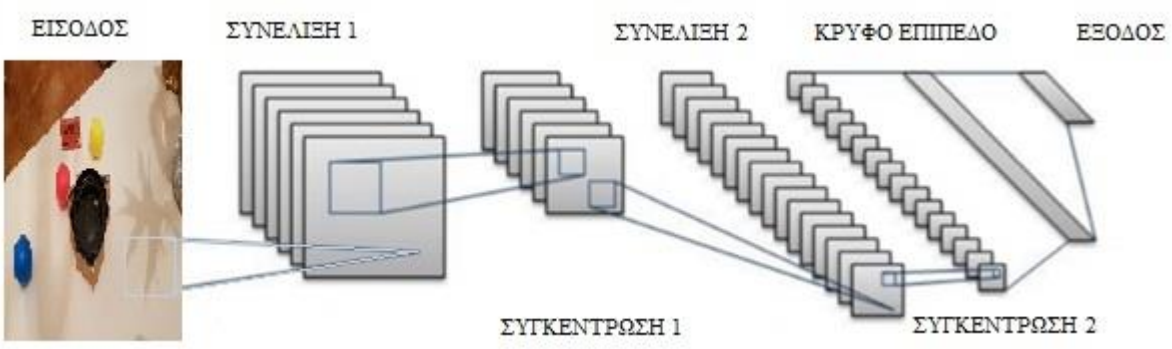
Αυτό που ουσιαστικά μας λέει ο τύπος είναι ότι, για ένα δεδομένο ερώτημα q, υπολογίζουμε το αντίστοιχο AP και στη συνέχεια, ο μέσος όρος όλων αυτών των βαθμολογιών AP θα μας έδινε έναν μόνο αριθμό, που ονομάζεται mAP, ο οποίος ποσοτικοποιεί πόσο καλό είναι το μοντέλο μας στην εκτέλεση του ερωτήματος.

Κεφάλαιο 2

Ανάλυση μοντέλων ανίχνευσης αντικειμένων

CNN

Ας ξεκινήσουμε με την απλούστερη προσέγγιση βαθιάς μάθησης, και μια ευρέως χρησιμοποιούμενης, για τον εντοπισμό αντικειμένων σε εικόνες, Convolutional Neural Networks ή CNNs. Θα αναφερθούμε εν συντομία στις εσωτερικές λειτουργίες ενός CNN.



Εικόνα 6

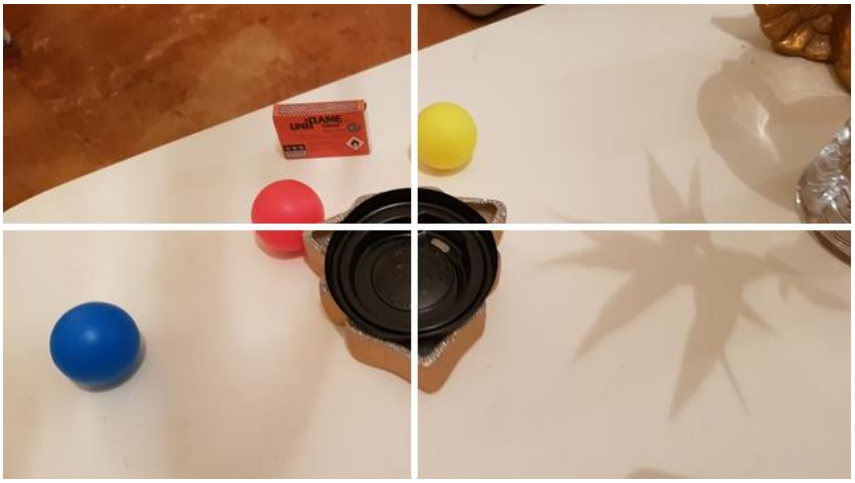
Παρατηρώντας τη παραπάνω εικόνα, εισάγεται μια εικόνα στο δίκτυο και στη συνέχεια αποστέλλεται μέσω διαφόρων συνεπειών και συγκεντρώσεων. Τέλος, λαμβάνεται η έξοδος με τη μορφή της κλάσης του αντικειμένου.

Για κάθε εικόνα εισόδου, λαμβάνεται μια αντίστοιχη κλάση ως έξοδος. Έτσι μπορούμε να χρησιμοποιήσουμε αυτήν την τεχνική για να ανιχνεύσουμε διάφορα αντικείμενα σε μια εικόνα. Ας δούμε πώς μπορούμε να λύσουμε ένα γενικό πρόβλημα ανίχνευσης αντικειμένων χρησιμοποιώντας ένα CNN.

- Πρώτα, παίρνουμε μια εικόνα ως είσοδο:

Εικόνα 7

- Στη συνέχεια διαιρούμε την εικόνα σε διάφορες περιοχές:



Εικόνα 8

- Έπειτα, θα θεωρήσουμε κάθε περιοχή ως ξεχωριστή εικόνα.
- Περνάμε όλες αυτές τις περιοχές (εικόνες) στο CNN και τις ταξινομούμε σε διάφορες κατηγορίες.
- Μόλις χωρίσουμε κάθε περιοχή στην αντίστοιχη κλάση, μπορούμε να συνδυάσουμε όλες αυτές τις περιοχές για να πάρουμε την αρχική εικόνα με τα αντικείμενα που εντοπίστηκαν.

Εικόνα 9

Το πρόβλημα με τη χρήση αυτής της προσέγγισης είναι ότι τα αντικείμενα στην εικόνα μπορούν να έχουν διαφορετικές αναλογίες διαστάσεων και χωρικές θέσεις. Για παράδειγμα, σε ορισμένες περιπτώσεις το αντικείμενο μπορεί να καλύπτει το μεγαλύτερο μέρος της εικόνας, ενώ σε άλλες το αντικείμενο μπορεί να καλύπτει μόνο ένα μικρό ποσοστό της εικόνας. Τα σχήματα των αντικειμένων μπορεί επίσης να είναι διαφορετικά (συμβαίνει πολύ σε πραγματικές περιπτώσεις χρήσης).

Ως αποτέλεσμα αυτών των παραγόντων, θα χρειαζόμασταν έναν πολύ μεγάλο αριθμό περιοχών με αποτέλεσμα ένα τεράστιο ποσό υπολογιστικού χρόνου. Για να λύσουμε αυτό το πρόβλημα και να μειώσουμε τον αριθμό των περιοχών, μπορούμε να χρησιμοποιήσουμε το CNN βάσει περιοχής, το οποίο επιλέγει τις περιοχές χρησιμοποιώντας μια μέθοδο πρότασης.

R-CNN

Αντί να επεξεργάζεται σε ένα τεράστιο αριθμό περιοχών, ο αλγόριθμος R-CNN προτείνει μια δέσμη κουτιών στην εικόνα που εισάγεται και ελέγχει εάν κάποιο από αυτά τα κουτιά περιέχει οποιοδήποτε αντικείμενο. Το R-CNN χρησιμοποιεί επιλεκτική αναζήτηση για να εξαγάγει αυτά τα πλαίσια από μια εικόνα (αυτά τα πλαίσια ονομάζονται περιοχές).

Υπάρχουν τέσσερις περιοχές που σχηματίζουν ένα αντικείμενο: ποικίλες κλίμακες, χρώματα, υφές και περίβλημα. Η επιλεκτική αναζήτηση προσδιορίζει αυτά τα μοτίβα στην εικόνα και βάσει αυτού, προτείνει διάφορες περιοχές. Συγκεκριμένα, παίρνουμε μία εικόνα ως είσοδο. Στη συνέχεια, η επιλεκτική αναζήτηση δημιουργεί υποδιαιρέσεις, έτσι ώστε να έχουμε

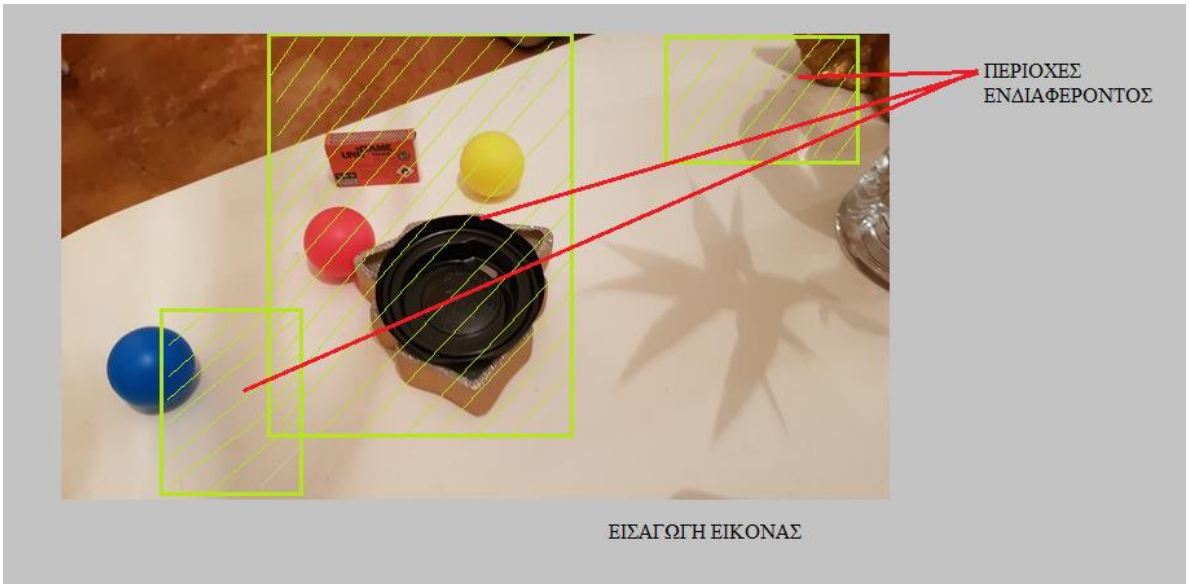
πολλές περιοχές από αυτήν την εικόνα. Η τεχνική συνδυάζει έπειτα τις παρόμοιες περιοχές για να σχηματίσει μια μεγαλύτερη περιοχή (με βάση την ομοιότητα χρωμάτων, την ομοιότητα υφής, την ομοιότητα μεγέθους και το σχήμα συμβατότητας). Τέλος, αυτές οι περιοχές παράγουν τις τελικές θέσεις αντικειμένων (Περιοχή ενδιαφέροντος). Αυτός είναι ο τρόπος λειτουργίας της επιλεκτικής αναζήτησης.

Ακολουθεί μια σύντομη περίληψη των βημάτων που ακολουθούνται στο R-CNN για τον εντοπισμό αντικειμένων:

- Πρώτα παίρνουμε ένα προεκπαιδευμένο συνελικτικό νευρωνικό δίκτυο.
- Στη συνέχεια, αυτό το μοντέλο επανεκπαιδεύεται. Εκπαιδεύουμε το τελευταίο επίπεδο του δικτύου με βάση τον αριθμό των τάξεων που πρέπει να εντοπιστούν.
- Το τρίτο βήμα είναι να αποκτήσουμε την περιοχή ενδιαφέροντος για κάθε εικόνα. Στη συνέχεια αναδιαμορφώνουμε όλες αυτές τις περιοχές έτσι ώστε να ταιριάζουν με το μέγεθος εισόδου CNN.
- Αφού αποκτήσουμε τις περιοχές, εκπαιδεύουμε το support-vector machine (SVM) για να ταξινομήσουμε αντικείμενα και φόντο. Για κάθε τάξη, εκπαιδεύουμε ένα δυαδικό SVM.
- Τέλος, εκπαιδεύουμε ένα μοντέλο γραμμικής παλινδρόμησης για τη δημιουργία αυστηρότερων πλαισίων οριοθέτησης για κάθε αναγνωρισμένο αντικείμενο στην εικόνα.

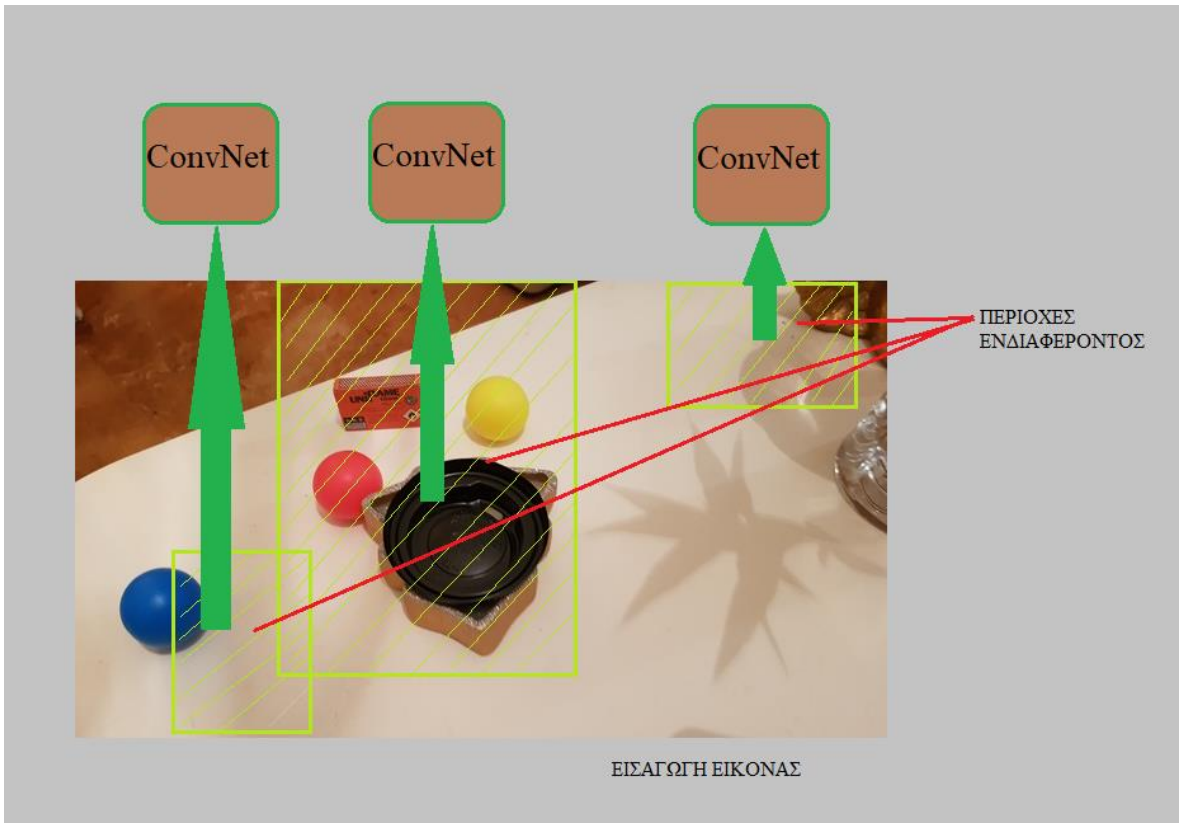
Παρακάτω παρατίθεται και ένα οπτικό παράδειγμα:

- Πρώτα, μια εικόνα λαμβάνεται ως είσοδος. Στο παράδειγμά μας, χρησιμοποιούμε την εικόνα 7.
- Στη συνέχεια, λαμβάνουμε τις περιοχές ενδιαφέροντος (ROI) χρησιμοποιώντας κάποια μέθοδο πρότασης (για παράδειγμα, επιλεκτική αναζήτηση όπως φαίνεται παρακάτω):



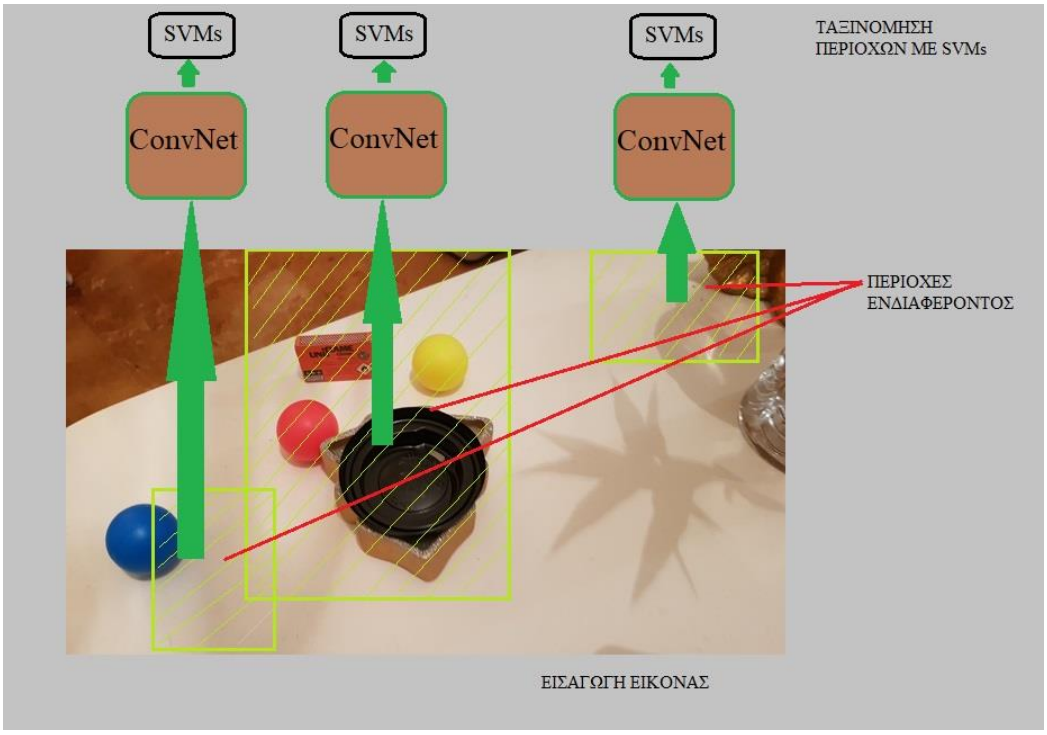
Εικόνα 10

- Όλες αυτές οι περιοχές αναδιαμορφώνονται στη συνέχεια σύμφωνα με την είσοδο του CNN και κάθε περιοχή μεταφέρεται στο ConvNet:



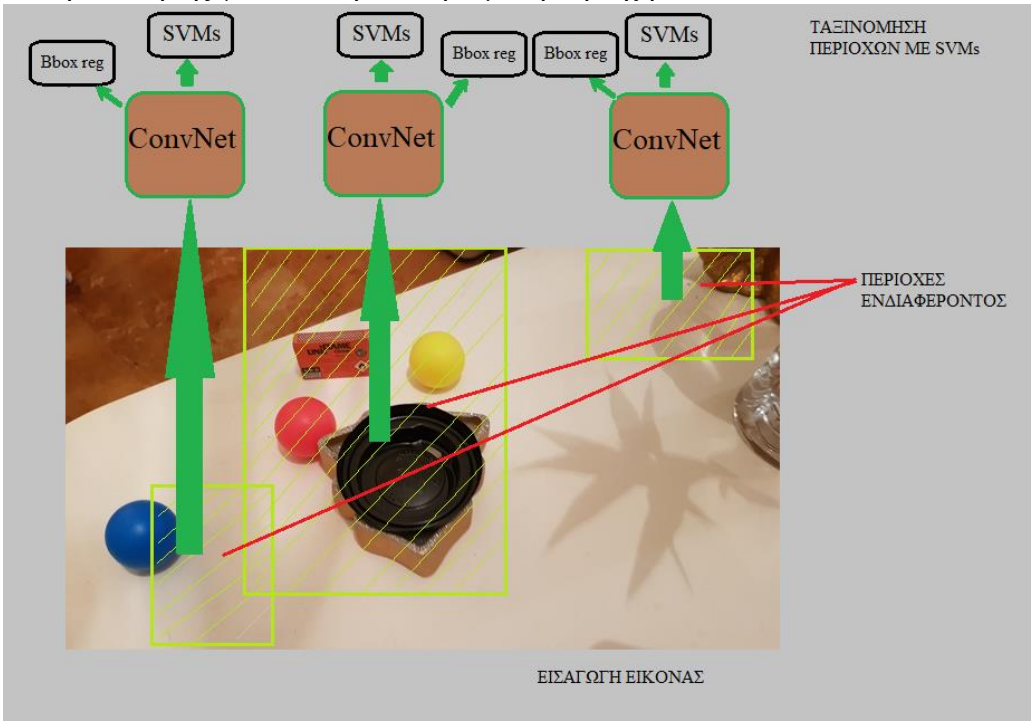
Εικόνα 11

- Στη συνέχεια, το CNN εξάγει χαρακτηριστικά για κάθε περιοχή και τα SVM χρησιμοποιούνται για να χωρίσουν αυτές τις περιοχές σε διαφορετικές κατηγορίες:



Εικόνα 12

- Τέλος, χρησιμοποιείται μια παλινδρόμηση οριοθέτησης (Bbox reg) για την πρόβλεψη των ορίων οριοθέτησης για κάθε προσδιορισμένη περιοχή:



Εικόνα 13

Με λίγα λόγια, αυτός είναι ο τρόπος με τον οποίο ένα R-CNN μας βοηθά να εντοπίζουμε αντικείμενα.

Προβλήματα με το R-CNN

Μέχρι στιγμής, έχουμε δει πώς το R-CNN μπορεί να είναι χρήσιμο για την ανίχνευση αντικειμένων. Αλλά αυτή η τεχνική έρχεται με τους δικούς της περιορισμούς. Η εκπαίδευση ενός μοντέλου R-CNN είναι ακριβή και αργή, χάρη στα παρακάτω βήματα:

- Εξαγωγή 2.000 περιοχών για κάθε εικόνα βάσει επιλεκτικής αναζήτησης.
- Εξαγωγή λειτουργιών χρησιμοποιώντας CNN για κάθε περιοχή εικόνας. Ας υποθέσουμε ότι έχουμε εικόνες N, τότε ο αριθμός των δυνατοτήτων CNN θα είναι N * 2.000.
- Η όλη διαδικασία ανίχνευσης αντικειμένων που χρησιμοποιεί το R-CNN έχει τρία μοντέλα:

1. CNN για εξαγωγή χαρακτηριστικών.
2. Γραμμικός ταξινομητής SVM για την αναγνώριση αντικειμένων.
3. Μοντέλο παλινδρόμησης για σύσφιξη των κουτιών οριοθέτησης.

Όλες αυτές οι διαδικασίες συνδυάζονται για να κάνουν το R-CNN πολύ αργό. Χρειάζονται περίπου 40-50 δευτερόλεπτα για να κάνουμε προβλέψεις για κάθε νέα εικόνα, κάτι που ουσιαστικά καθιστά το μοντέλο δυσκίνητο και πρακτικά αδύνατο να δημιουργηθεί όταν αντιμετωπίζει ένα τεράστιο σύνολο δεδομένων.

Υπάρχει όμως, μια άλλη τεχνική ανίχνευσης αντικειμένων που διορθώνει τους περισσότερους από τους περιορισμούς που είδαμε στο R-CNN.

Fast R-CNN

Για να μειώσουμε τον χρόνο υπολογισμού που χρειάζεται συνήθως ένας αλγόριθμος R-CNN, αντί να εκτελέσουμε CNN 2.000 φορές ανά εικόνα, μπορούμε να το εκτελέσουμε μόνο μία φορά ανά εικόνα και να λάβουμε όλες τις περιοχές ενδιαφέροντος (περιοχές που περιέχουν κάποιο αντικείμενο).

Στο Fast R-CNN, τροφοδοτούμε την εικόνα εισόδου στο CNN, το οποίο με τη σειρά του δημιουργεί τους χάρτες χαρακτηριστικών. Χρησιμοποιώντας αυτούς τους χάρτες, εξάγονται οι περιοχές των προτάσεων. Στη συνέχεια, χρησιμοποιούμε ένα στρώμα συγκέντρωσης RoI για να αναδιαμορφώσουμε όλες τις προτεινόμενες περιοχές σε σταθερό μέγεθος, έτσι ώστε να μπορεί να τροφοδοτηθεί σε ένα πλήρως συνδεδεμένο δίκτυο.

Ας το αναλύσουμε σε βήματα για την απλοποίηση της έννοιας:

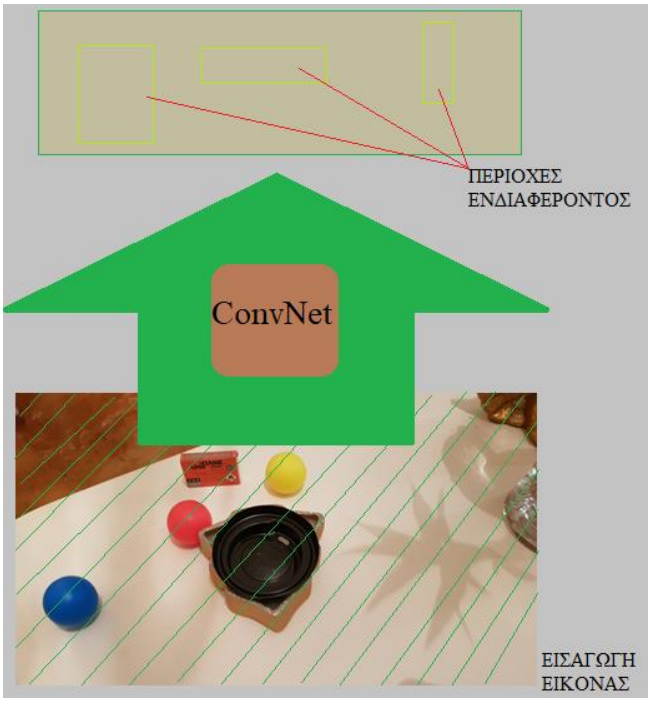
- Όπως και με τις δύο προηγούμενες τεχνικές, λαμβάνουμε μια εικόνα ως είσοδο.
- Αυτή η εικόνα μεταφέρεται σε ένα ConvNet το οποίο με τη σειρά του δημιουργεί τις περιοχές ενδιαφέροντος.
- Εφαρμόζεται ένα στρώμα συγκέντρωσης RoI σε όλες αυτές τις περιοχές για να τα αναδιαμορφώσουν σύμφωνα με την είσοδο του ConvNet. Στη συνέχεια, κάθε περιοχή μεταφέρεται σε ένα πλήρως συνδεδεμένο δίκτυο.
- Ένα στρώμα softmax χρησιμοποιείται πάνω από το πλήρως συνδεδεμένο δίκτυο με τάξεις εξόδου. Μαζί με το στρώμα softmax, ένα γραμμικό στρώμα παλινδρόμησης χρησιμοποιείται επίσης παράλληλα για την παραγωγή συντεταγμένων κουτιού οριοθέτησης για προβλεπόμενες κατηγορίες.

Έτσι, αντί να χρησιμοποιεί τρία διαφορετικά μοντέλα (όπως στο R-CNN), το Fast R-CNN χρησιμοποιεί ένα μόνο μοντέλο που εξάγει χαρακτηριστικά από τις περιοχές, τα χωρίζει σε διαφορετικές κατηγορίες και επιστρέφει τα κουτιά ορίων για τις αναγνωρισμένες κλάσεις ταυτόχρονα.

Για να το αναλύσουμε ακόμη περισσότερο, θα απεικονίσουμε κάθε βήμα για να προσθέσουμε μια πρακτική γωνία στην εξήγηση.

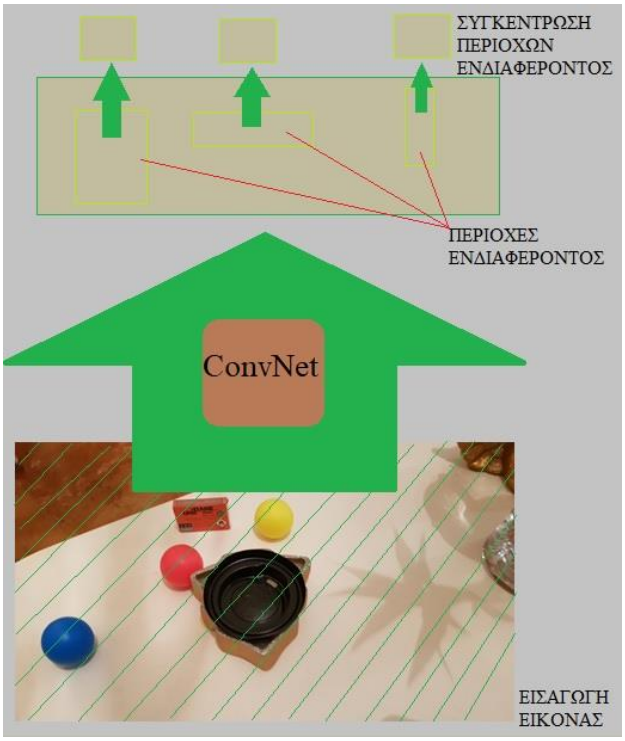
- Ακολουθούμε το πλέον γνωστό βήμα της λήψης μιας εικόνας ως είσοδος. Στο παράδειγμά μας, χρησιμοποιούμε την εικόνα 7.

- Αυτή η εικόνα μεταβιβάζεται σε ένα ConvNet το οποίο επιστρέφει ανάλογα την περιοχή ενδιαφέροντος:



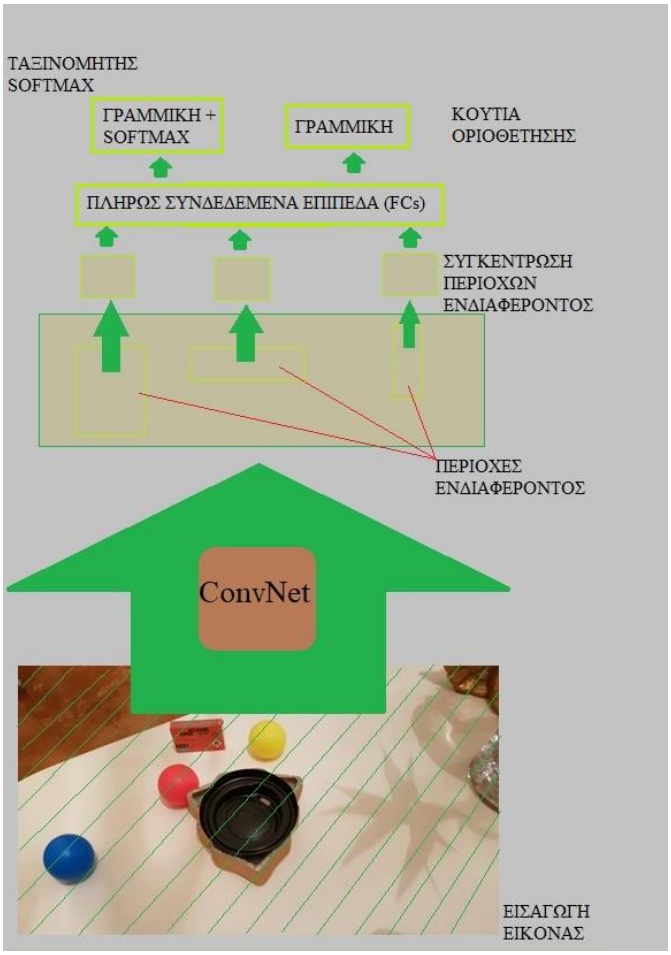
Εικόνα 14

- Στη συνέχεια εφαρμόζουμε το στρώμα συγκέντρωσης RoI στις εξαγόμενες περιοχές ενδιαφέροντος για να βεβαιωθούμε ότι όλες οι περιοχές έχουν το ίδιο μέγεθος:



Εικόνα 15

- Τέλος, αυτές οι περιοχές μεταβιβάζονται σε ένα πλήρως συνδεδεμένο δίκτυο που τις ταξινομεί, καθώς και επιστρέφει τα οριακά κουτιά χρησιμοποιώντας στρώματα softmax και γραμμικής παλινδρόμησης ταυτόχρονα:



Εικόνα 16

Αυτός είναι ο τρόπος με τον οποίο το Fast R-CNN επιλύει δύο σημαντικά ζητήματα του R-CNN, δηλαδή, μεταφέροντας ένα αντί για 2.000 περιοχές ανά εικόνα στο ConvNet και χρησιμοποιώντας ένα αντί για τρία διαφορετικά μοντέλα για εξαγωγή χαρακτηριστικών, ταξινόμηση και δημιουργία πλαισίων οριοθέτησης.

Προβλήματα με το Fast R-CNN

Αλλά ακόμη και το Fast R-CNN έχει ορισμένες προβληματικές περιοχές. Χρησιμοποιεί επίσης επιλεκτική αναζήτηση ως μέθοδο πρότασης για να βρει τις περιοχές ενδιαφέροντος, που είναι μια αργή και χρονοβόρα διαδικασία. Χρειάζονται περίπου 2 δευτερόλεπτα ανά εικόνα για τον εντοπισμό αντικειμένων, κάτι που είναι πολύ καλύτερο σε σύγκριση με το R-CNN. Αλλά

όταν εξετάζουμε μεγάλα σύνολα δεδομένων πραγματικής ζωής, τότε ακόμη και ένα Fast R-CNN δεν φαίνεται πλέον τόσο γρήγορο.

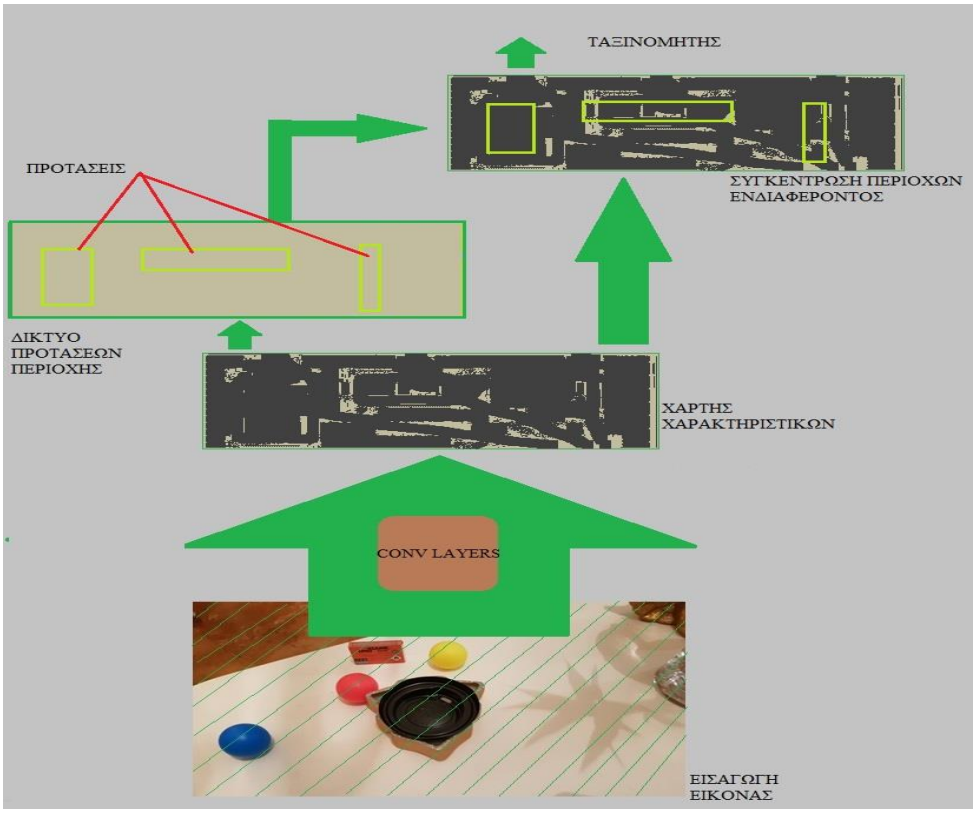
Υπάρχει όμως ένας άλλος αλγόριθμος ανίχνευσης αντικειμένων που υπερασπίζεται το Fast R-CNN.

Faster R-CNN

Το Faster R-CNN είναι η τροποποιημένη έκδοση του Fast R-CNN. Η κύρια διαφορά μεταξύ τους είναι ότι το Fast R-CNN χρησιμοποιεί επιλεκτική αναζήτηση για τη δημιουργία περιοχών ενδιαφέροντος, ενώ το Faster R-CNN χρησιμοποιεί το "Region Proposal Network", γνωστό και ως RPN. Το RPN λαμβάνει τους χάρτες χαρακτηριστικών εικόνας ως είσοδο και δημιουργεί ένα σύνολο προτάσεων αντικειμένων, καθένα με βαθμολογία αντικειμενικότητας ως έξοδο.

Τα παρακάτω βήματα ακολουθούνται συνήθως σε μια προσέγγιση του Faster R-CNN:

- Παίρνουμε μια εικόνα ως είσοδο και τη μεταδίδουμε στο ConvNet που επιστρέφει το χάρτη δυνατοτήτων για αυτήν την εικόνα.
- Το δίκτυο προτάσεων περιοχής εφαρμόζεται σε αυτούς τους χάρτες χαρακτηριστικών. Αυτό επιστρέφει τις προτάσεις αντικειμένου μαζί με τη βαθμολογία αντικειμενικότητάς τους.
- Εφαρμόζεται ένα στρώμα συγκέντρωσης RoI σε αυτές τις προτάσεις για να μειωθούν όλες οι προτάσεις στο ίδιο μέγεθος.
- Τέλος, οι προτάσεις περνούν σε ένα πλήρως συνδεδεμένο στρώμα που έχει ένα στρώμα softmax και ένα γραμμικό στρώμα παλινδρόμησης στην κορυφή του, για την ταξινόμηση και την έξοδο των ορίων οριοθέτησης για αντικείμενα.



Εικόνα 17

Εν συντομία θα αναφερθούμε πώς λειτουργεί αυτό το δίκτυο πρότασης περιοχής (RPN).

Αρχικά, το Faster R-CNN παίρνει τους χάρτες χαρακτηριστικών από το CNN και τους μεταφέρει στο Δίκτυο Πρότασης Περιφέρειας. Το RPN χρησιμοποιεί ένα συρόμενο παράθυρο πάνω από αυτούς τους χάρτες χαρακτηριστικών και σε κάθε παράθυρο δημιουργεί k κουτιά αγκύρωσης διαφορετικών σχημάτων και μεγεθών:

Εικόνα 18

Τα κουτιά αγκύρωσης είναι οριακά κουτιά σταθερού μεγέθους που τοποθετούνται σε όλη την εικόνα και έχουν διαφορετικά σχήματα και μεγέθη. Για κάθε κουτί αγκύρωσης, το RPN προβλέπει δύο πράγματα:

- Το πρώτο είναι η πιθανότητα ότι ένα κουτί αγκύρωσης είναι αντικείμενο (δεν λαμβάνει υπόψη σε ποια κατηγορία ανήκει το αντικείμενο).
- Το δεύτερο είναι το παλινδρομικό κουτί οριοθέτησης για ρύθμιση των κουτιών αγκύρωσης ώστε να ταιριάζει καλύτερα στο αντικείμενο.

Έχουμε τώρα οριακά κουτιά διαφορετικών σχημάτων και μεγεθών που μεταφέρονται στο στρώμα ομαδοποίησης RoI. Είναι πιθανό ότι μετά το βήμα RPN, υπάρχουν προτάσεις χωρίς να έχουν ανατεθεί τάξεις. Μπορούμε να πάρουμε κάθε πρόταση και να την περικόψουμε έτσι ώστε κάθε πρόταση να περιέχει ένα αντικείμενο. Αυτό κάνει το στρώμα συγκέντρωσης RoI. Εξάγει σταθερούς μεγέθους χάρτες χαρακτηριστικών για κάθε κουτί αγκύρωσης:

Στη συνέχεια, αυτοί οι χάρτες χαρακτηριστικών μεταφέρονται σε ένα πλήρως συνδεδεμένο επίπεδο που έχει ένα softmax και ένα επίπεδο γραμμικής παλινδρόμησης. Τελικά ταξινομεί το αντικείμενο και προβλέπει τα πλαίσια οριοθέτησης για τα αναγνωρισμένα αντικείμενα.

Προβλήματα με Faster R-CNN

Όλοι οι αλγόριθμοι ανίχνευσης αντικειμένων που έχουμε συζητήσει μέχρι στιγμής χρησιμοποιούν περιοχές για τον προσδιορισμό των αντικειμένων. Το δίκτυο δεν εξετάζει την πλήρη εικόνα με μία κίνηση, αλλά εστιάζει σε τμήματα της εικόνας διαδοχικά. Αυτό δημιουργεί δύο επιπλοκές:

- Ο αλγόριθμος απαιτεί πολλά περάσματα από μία μόνο εικόνα για την εξαγωγή όλων των αντικειμένων.
- Δεδομένου ότι υπάρχουν διαφορετικά συστήματα που λειτουργούν το ένα μετά το άλλο, η απόδοση των συστημάτων πιο μπροστά εξαρτάται από την απόδοση των προηγούμενων συστημάτων.

YOLO

Το YOLO είναι ένα σύστημα ανίχνευσης αντικειμένων που στοχεύει σε επεξεργασία σε πραγματικό χρόνο.

Η οικογένεια τεχνικών R-CNN που έχουμε προαναφέρει χρησιμοποιεί κυρίως περιοχές για να εντοπίσει τα αντικείμενα μέσα στην εικόνα. Το δίκτυο δεν βλέπει ολόκληρη την εικόνα, μόνο στα τμήματα των εικόνων που έχουν περισσότερες πιθανότητες να περιέχουν ένα αντικείμενο.

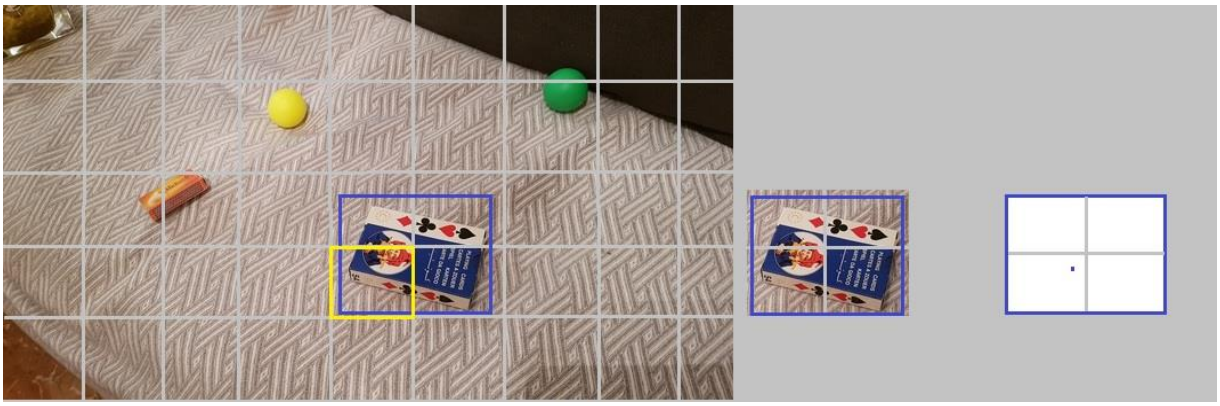
Από την άλλη πλευρά, το πλαίσιο YOLO, ασχολείται με την ανίχνευση αντικειμένων με διαφορετικό τρόπο. Παίρνει ολόκληρη την εικόνα σε μία παρουσία και προβλέπει τις συντεταγμένες πλαισίου οριοθέτησης και τις πιθανότητες κλάσης για αυτά τα πλαίσια. Το μεγαλύτερο πλεονέκτημα της χρήσης του YOLO είναι η εξαιρετική του ταχύτητα. Είναι απίστευτα γρήγορο και μπορεί να επεξεργαστεί 45 καρέ ανά δευτερόλεπτο. Το YOLO κατανοεί επίσης τη γενικευμένη αναπαράσταση αντικειμένων.

Αυτός είναι ένας από τους καλύτερους αλγόριθμους για την ανίχνευση αντικειμένων και έχει δείξει μια σχετικά παρόμοια απόδοση με τους αλγόριθμους R-CNN.

Παρακάτω θα αναλύσουμε τη διαδικασία που πραγματοποιεί το YOLO:

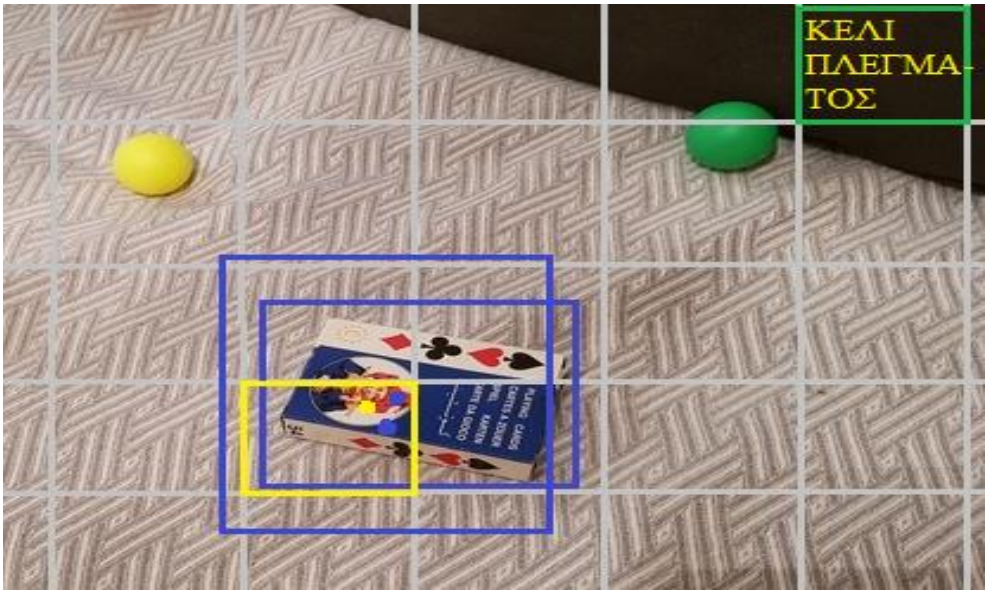
Κελί πλέγματος

Το YOLO διαιρεί την εικόνα εισόδου σε πλέγμα $S * S$. Κάθε κελί πλέγματος προβλέπει μόνο ένα αντικείμενο. Για παράδειγμα, το κίτρινο κελί πλέγματος παρακάτω προσπαθεί να προβλέψει το αντικείμενο "τράπουλα" του οποίου το κέντρο (η μπλε κουκκίδα) πέφτει μέσα στο κελί πλέγματος.



Εικόνα 19

Κάθε κελί πλέγματος προβλέπει έναν σταθερό αριθμό κουτιών ορίου. Σε αυτό το παράδειγμα, το κίτρινο κελί πλέγματος κάνει δύο προβλέψεις κουτιού ορίου (μπλε κουτιά) για να εντοπίσει πού βρίσκεται η τράπουλα.



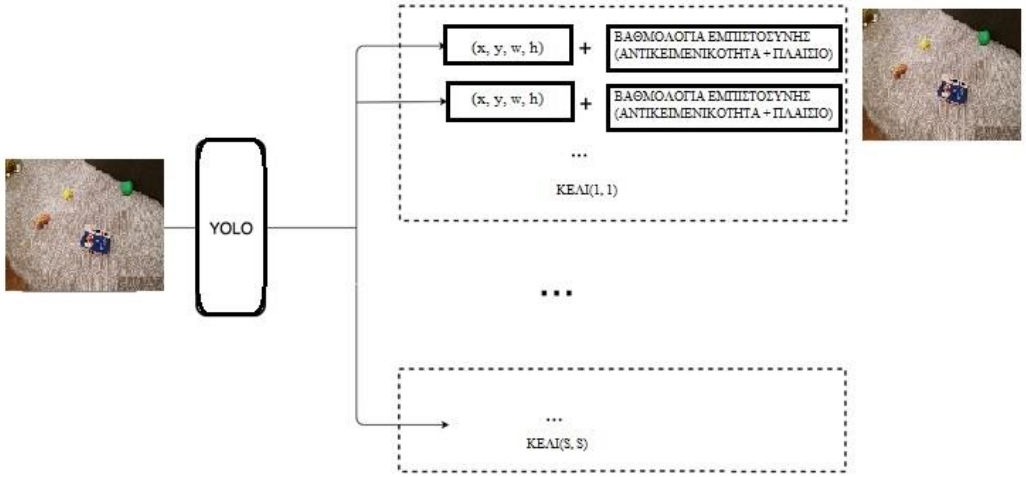
Εικόνα 20

Ωστόσο, ο κανόνας ενός αντικειμένου περιορίζει το πόσο κοντά μπορεί να είναι τα αντικείμενα που εντοπίζονται. Για αυτό, το YOLO έχει ορισμένους περιορισμούς στο πόσο κοντά είναι τα αντικείμενα.

Για κάθε κελί πλέγματος,

- προβλέπει B όρια και κάθε κουτί έχει ένα σκορ εμπιστοσύνης,
- ανιχνεύει ένα αντικείμενο μόνο ανεξάρτητα από τον αριθμό των κουτιών B,
- προβλέπει C πιθανότητες τάξης υπό όρους (μία ανά τάξη για την ομοιότητα της κλάσης αντικειμένων).

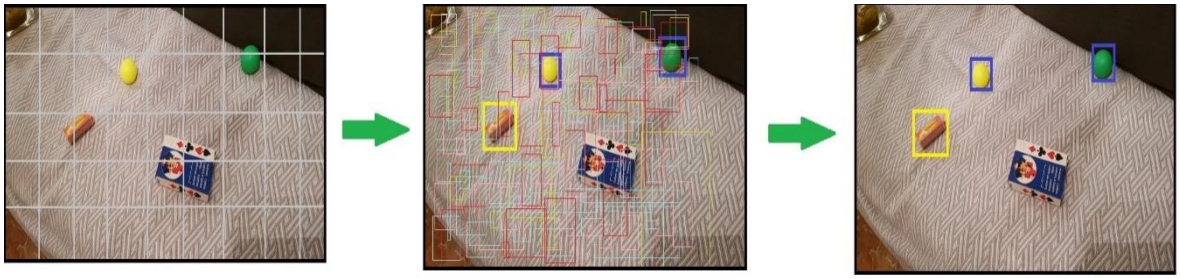
Για να αξιολογήσει το PASCAL VOC, το YOLO χρησιμοποιεί πλέγματα $7 * 7$ ($S * S$), 2 κουτιά ορίων (B) και 20 τάξεις (C).



Εικόνα 21

Ας δούμε περισσότερες λεπτομέρειες. Κάθε πλαίσιο ορίου περιέχει 5 στοιχεία: (x, y, w, h) και βαθμολογία εμπιστοσύνης κουτιού. Η βαθμολογία εμπιστοσύνης αντικατοπτρίζει το πόσο πιθανό είναι το πλαίσιο να περιέχει ένα αντικείμενο (αντικειμενικότητα) και πόσο ακριβής είναι το πλαίσιο ορίου. Ομαλοποιούμε το πλάτος του πλαισίου οριοθέτησης w και το ύψος h από το πλάτος και το ύψος της εικόνας, x και y είναι αντισταθμίσεις στο αντίστοιχο κελί. Επομένως, τα x, y, w και h είναι όλα μεταξύ 0 και 1. Κάθε κελί έχει 20 πιθανότητες κατηγορίας υπό όρους. Η πιθανότητα κλάσης υπό όρους είναι η πιθανότητα ότι το αντικείμενο που ανιχνεύεται ανήκει σε μια συγκεκριμένη κατηγορία (μία πιθανότητα ανά κατηγορία για κάθε κελί). Έτσι, η πρόβλεψη του YOLO έχει σχήμα $(\mathrm{S}, \mathrm{S}, \mathrm{B} * 5 + \mathrm{C}) = (7{,}7{,}2 * 5 + 20) = (7{,}7{,}30)$.

Η κύρια ιδέα του YOLO είναι η δημιουργία ενός δικτύου CNN για την πρόβλεψη ενός τανυστή (7, 7, 30). Χρησιμοποιεί ένα δίκτυο CNN για να μειώσει τη χωρική διάσταση σε $7 * 7$ με 1024 κανάλια εξόδου σε κάθε τοποθεσία. Το YOLO εκτελεί μια γραμμική παλινδρόμηση χρησιμοποιώντας δύο πλήρως συνδεδεμένα στρώματα για να κάνει προβλέψεις πλαισίου ορίου $7 * 7 * 2$ (το μεσαίο εικονίδιο της εικόνας 22). Για να κάνουμε μια τελική πρόβλεψη, διατηρούμε εκείνες με υψηλές βαθμολογίες εμπιστοσύνης κουτιού (μεγαλύτερες από 0.25) ως τις τελικές προβλέψεις μας.



Εικόνα 22

Η βαθμολογία εμπιστοσύνης τάξης για κάθε πλαίσιο πρόβλεψης υπολογίζεται ως:

$$\text{class confidence score} = \text{box confidence score} * \text{conditional class probability} \quad (1.14)$$

Μετρά την εμπιστοσύνη τόσο στην ταξινόμηση όσο και στον εντοπισμό (όπου βρίσκεται ένα αντικείμενο). Μπορούμε να συνδυάσουμε εύκολα αυτούς τους όρους βαθμολογίας και πιθανότητας.

Ακολουθούν οι μαθηματικοί ορισμοί:

$\text{box confidence score} \equiv P_r(\text{object}) \cdot \text{IoU}$

$\text{conditional class probability} \equiv P_r(\text{class}_i|\text{object})$

$\text{class confidence score} \equiv P_r(\text{class}_i) \cdot \text{IoU}$

$= \text{box confidence score} * \text{conditional class probability}$

Όπου:

$P_r(\text{object})$ είναι η πιθανότητα το πλαίσιο να περιέχει ένα αντικείμενο.

IoU (intersection over union) μεταξύ του προβλεπόμενου κουτιού και της αλήθειας.

$P_r(\text{class}_i|\text{object})$ είναι η πιθανότητα το αντικείμενο να ανήκει στο class_i δεδομένου ότι ένα αντικείμενο είναι παρόν.

$P_r(\text{class}_i)$ είναι η πιθανότητα το αντικείμενο να ανήκει στο class_i.

Σχεδιασμός δικτύου

Το YOLO έχει 24 συνελικτικά στρώματα ακολουθούμενο από 2 πλήρως συνδεδεμένα επίπεδα (FC) που βοηθά στη χαρτογράφηση της αναπαράστασης μεταξύ της εισόδου και της εξόδου. Ορισμένα στρώματα συνελεύσεων χρησιμοποιούν εναλλακτικά επίπεδα μείωσης $1 * 1$

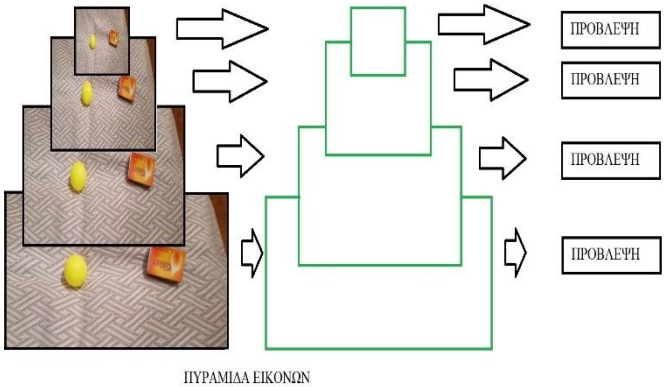
για να μειώσουν το βάθος των χαρτών χαρακτηριστικών. Για το τελευταίο επίπεδο συνελεύσεων, εξάγει έναν τανυστή με σχήμα (7, 7, 1024). Ο τανυστής ισοπεδώνεται. Χρησιμοποιώντας 2 πλήρως συνδεδεμένα επίπεδα ως μορφή γραμμικής παλινδρόμησης, εξάγει παραμέτρους $7 * 7 * 30$ και στη συνέχεια αναδιαμορφώνεται σε (7, 7, 30), δηλαδή 2 προβλέψεις κουτιού ορίου ανά τοποθεσία.

Οφέλη του YOLO

- Γρήγορο. Καλό για επεξεργασία σε πραγματικό χρόνο.
- Οι προβλέψεις (θέσεις αντικειμένων και τάξεις) γίνονται από ένα μόνο δίκτυο. Μπορεί να εκπαιδευτεί από άκρο σε άκρο για να βελτιώσει την ακρίβεια.
- Το YOLO είναι πιο γενικευμένο. Ξεπερνά άλλες μεθόδους κατά τη γενίκευση, από φυσικές εικόνες σε άλλους τομείς όπως το έργο τέχνης.
- Οι μέθοδοι πρότασης περιοχής περιορίζουν τον ταξινομητή στη συγκεκριμένη περιοχή. Το YOLO έχει πρόσβαση σε ολόκληρη την εικόνα κατά την πρόβλεψη των ορίων. Με το πρόσθετο πλαίσιο, το YOLO εμφανίζει λιγότερα ψευδώς θετικά σε περιοχές φόντου.
- Το YOLO ανιχνεύει ένα αντικείμενο ανά κελί πλέγματος. Επιβάλλει τη χωρική ποικιλομορφία στην πραγματοποίηση προβλέψεων.

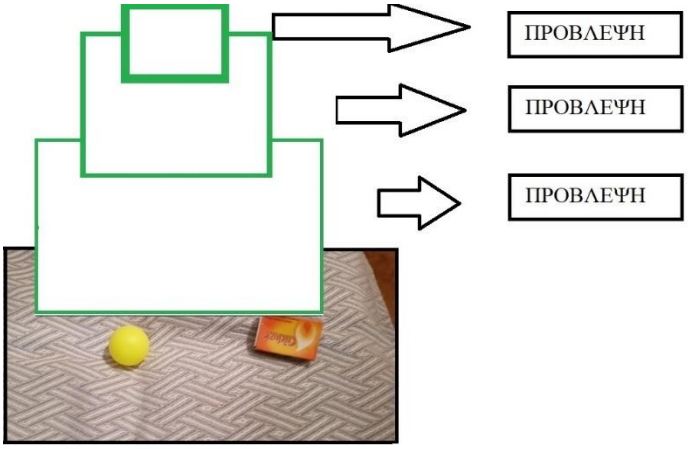
Feature Pyramid Networks for object detection (FPN)

Η ανίχνευση αντικειμένων σε διαφορετικές κλίμακες είναι δύσκολη, ιδίως για μικρά αντικείμενα. Μπορούμε να χρησιμοποιήσουμε μια πυραμίδα της ίδιας εικόνας σε διαφορετική κλίμακα για να εντοπίσουμε αντικείμενα (η παρακάτω εικόνα).



Εικόνα 23

Ωστόσο, η επεξεργασία εικόνων πολλαπλής κλίμακας είναι χρονοβόρα και η ζήτηση μνήμης είναι πολύ υψηλή για να εκπαιδευτεί ταυτόχρονα από άκρο σε άκρο. Ως εκ τούτου, μπορούμε να το χρησιμοποιήσουμε μόνο για να προωθήσουμε την ακρίβεια όσο το δυνατόν υψηλότερα, ιδίως για διαγωνισμούς, όταν η ταχύτητα δεν αποτελεί πρόβλημα. Εναλλακτικά, δημιουργούμε μια πυραμίδα χαρακτηριστικών και τις χρησιμοποιούμε για ανίχνευση αντικειμένων (η ακόλουθη εικόνα). Ωστόσο, οι χάρτες χαρακτηριστικών πλησιάζουν το επίπεδο εικόνας που αποτελείται από δομές χαμηλού επιπέδου που δεν είναι αποτελεσματικές για την ακριβή ανίχνευση αντικειμένων.



ΠΥΡΑΜΙΔΑ ΧΑΡΤΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ

Εικόνα 24

Το Feature Pyramid Network (FPN) είναι ένα εργαλείο εξαγωγής χαρακτηριστικών σχεδιασμένο για μια τέτοια ιδέα πυραμίδας με γνώμονα την ακρίβεια και την ταχύτητα. Αντικαθιστά τον εξολκέα χαρακτηριστικών ανιχνευτών όπως το Faster R-CNN και δημιουργεί πολλαπλά επίπεδα χαρτών χαρακτηριστικών (χάρτες πολλαπλών κλιμάκων) με πληροφορίες καλύτερης ποιότητας από την κανονική πυραμίδα χαρακτηριστικών για την ανίχνευση αντικειμένων.

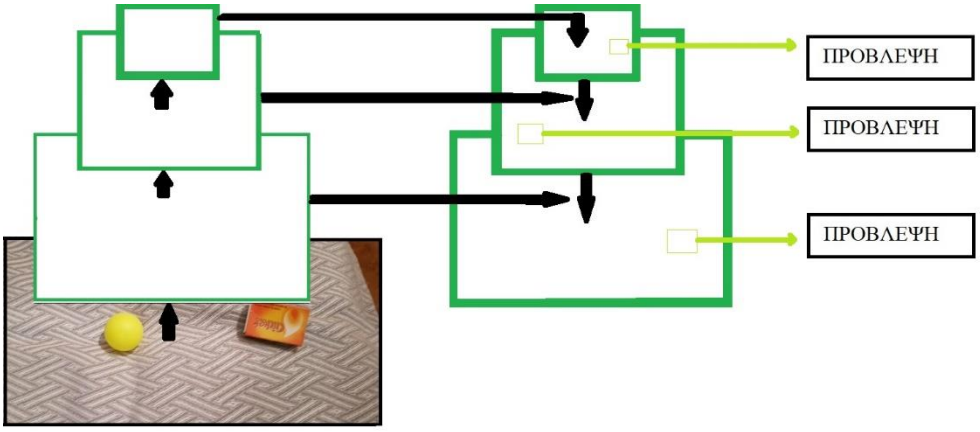
Ροή δεδομένων

Το FPN αποτελείται από μια διαδρομή από κάτω προς τα πάνω και από πάνω προς τα κάτω. Η διαδρομή από κάτω προς τα πάνω είναι το συνηθισμένο συνελικτικό δίκτυο για εξαγωγή χαρακτηριστικών. Καθώς ανεβαίνουμε, η χωρική ανάλυση μειώνεται. Με την ανίχνευση περισσότερων δομών υψηλού επιπέδου, η σημασιολογική τιμή για κάθε στρώμα αυξάνεται.

Το SSD πραγματοποιεί εντοπισμό από πολλούς χάρτες χαρακτηριστικών. Ωστόσο, τα κάτω επίπεδα δεν επιλέγονται για ανίχνευση αντικειμένων. Είναι σε υψηλή ανάλυση, αλλά η σημασιολογική τιμή δεν είναι αρκετά υψηλή για να δικαιολογήσει τη χρήση του καθώς η

επιβράδυνση της ταχύτητας είναι σημαντική. Έτσι, το SSD χρησιμοποιεί μόνο ανώτερα επίπεδα για ανίχνευση και επομένως αποδίδει πολύ χειρότερα για μικρά αντικείμενα.

Το FPN παρέχει ένα μονοπάτι από πάνω προς τα κάτω για την κατασκευή επιπέδων υψηλότερης ανάλυσης από ένα πλούσιο σε σημασιολογικό επίπεδο.



Εικόνα 25: Μοντέλο FPN.

Ενώ τα ανακατασκευασμένα στρώματα είναι σημασιολογικά ισχυρά, αλλά οι θέσεις των αντικειμένων δεν είναι ακριβείς μετά από όλες τις δειγματοληψίες της βάσης και τις δειγματοληψίες από τη κορυφή. Προσθέτουμε πλευρικές συνδέσεις μεταξύ ανακατασκευασμένων επιπέδων και των αντίστοιχων χαρτών χαρακτηριστικών για να βοηθήσουμε τον ανιχνευτή να προβλέψει την καλύτερη θέση. Λειτουργεί επίσης ως παράλειψη συνδέσεων για να διευκολύνει την εκπαίδευση (παρόμοιο με αυτό που κάνει το ResNet).

MobileNet

Το μοντέλο MobileNet βασίζεται σε βάθος διαχωρίσιμες (περιελιγμούς) συνελεύσεις που είναι μια μορφή παραγοντοποιημένων συνεπειών. Αυτά παραγοντοποιούν μια τυπική συνέλιξη σε μια κατά βάθος συνέλιξη και μια διάκριση $1 * 1$ που ονομάζεται κατάταξη.

Για το MobileNet, η σε βάθος ανάλυση εφαρμόζει ένα μόνο φίλτρο σε κάθε κανάλι εισόδου. Στη συνέχεια, η περιστροφική κατάκλιση εφαρμόζει μια συνέλιξη $1 * 1$ για να συνδυάσει τις εξόδους της βάσης.

Μια τυπική συνέλιξη και με τα δύο φίλτρα συνδυάζει τις εισόδους σε ένα νέο σύνολο εξόδων σε ένα βήμα. Το βάθος που μπορεί να διαχωριστεί χωρίζει αυτό σε δύο στρώματα. Ένα ξεχωριστό στρώμα για φιλτράρισμα και ένα ξεχωριστό στρώμα για συνδυασμό. Αυτή η παραγοντοποίηση έχει ως αποτέλεσμα τη δραστική μείωση του υπολογισμού και του μεγέθους του μοντέλου.

MobileNet-SSD

Η αρχιτεκτονική Single Shot Detector (SSD) είναι ένα μοναδικό δίκτυο συνελεύσεων που μαθαίνει να προβλέπει θέσεις οριοθέτησης και ταξινόμησης αυτών των τοποθεσιών με ένα πέρασμα. Ως εκ τούτου, το SSD μπορεί να εκπαιδευτεί από άκρο σε άκρο. Το δίκτυο SSD αποτελείται από βασική αρχιτεκτονική (MobileNet σε αυτήν την περίπτωση) ακολουθούμενη από διάφορα επίπεδα συνδιαλλαγής:

Το SSD λειτουργεί σε χάρτες χαρακτηριστικών για να εντοπίσει τη θέση των πλαισίων οριοθέτησης. Αξίζει να σημειωθεί ότι ένας χάρτης χαρακτηριστικών έχει το μέγεθος $\mathrm{Df} * \mathrm{Df} * \mathrm{M}$. Για κάθε τοποθεσία χάρτη χαρακτηριστικών, προβλέπονται κουτιά οριοθέτησης k. Κάθε κουτί οριοθέτησης φέρει μαζί του τις ακόλουθες πληροφορίες:

- 4 θέσεις μετατόπισης πλαισίου οριοθέτησης γωνίας (cx, cy, w, h)
- Πιθανότητες κλάσης C (c1, c2, …, cp)

Το SSD δεν προβλέπει το σχήμα του κουτιού, αλλά ακριβώς πού βρίσκεται το κουτί. Τα κιβώτια οριοθέτησης k έχουν ένα προκαθορισμένο σχήμα. Τα σχήματα έχουν οριστεί πριν από την πραγματική εκπαίδευση.

Απώλειες στο MobileNet-SSD

Με το τελικό σετ αντιστοιχισμένων κουτιών, μπορούμε να υπολογίσουμε την απώλεια ως εξής:

$$\mathrm{L} = \frac{1}{\mathrm{N}}(\mathrm{L\ class} + \mathrm{L\ box}) \qquad (1.15)$$

Εδώ, N είναι ο συνολικός αριθμός αντιστοιχισμένων κουτιών. Η κλάση L είναι η απώλεια softmax για ταξινόμηση και το $\mathrm{L\ box}$ είναι η ομαλή απώλεια L1 που αντιπροσωπεύει το σφάλμα των αντίστοιχων κουτιών. Η ομαλή απώλεια L1 είναι μια τροποποίηση της απώλειας L1 που είναι πιο ισχυρή για τα outliers. Σε περίπτωση που το N είναι 0, η απώλεια ορίζεται και στο 0.

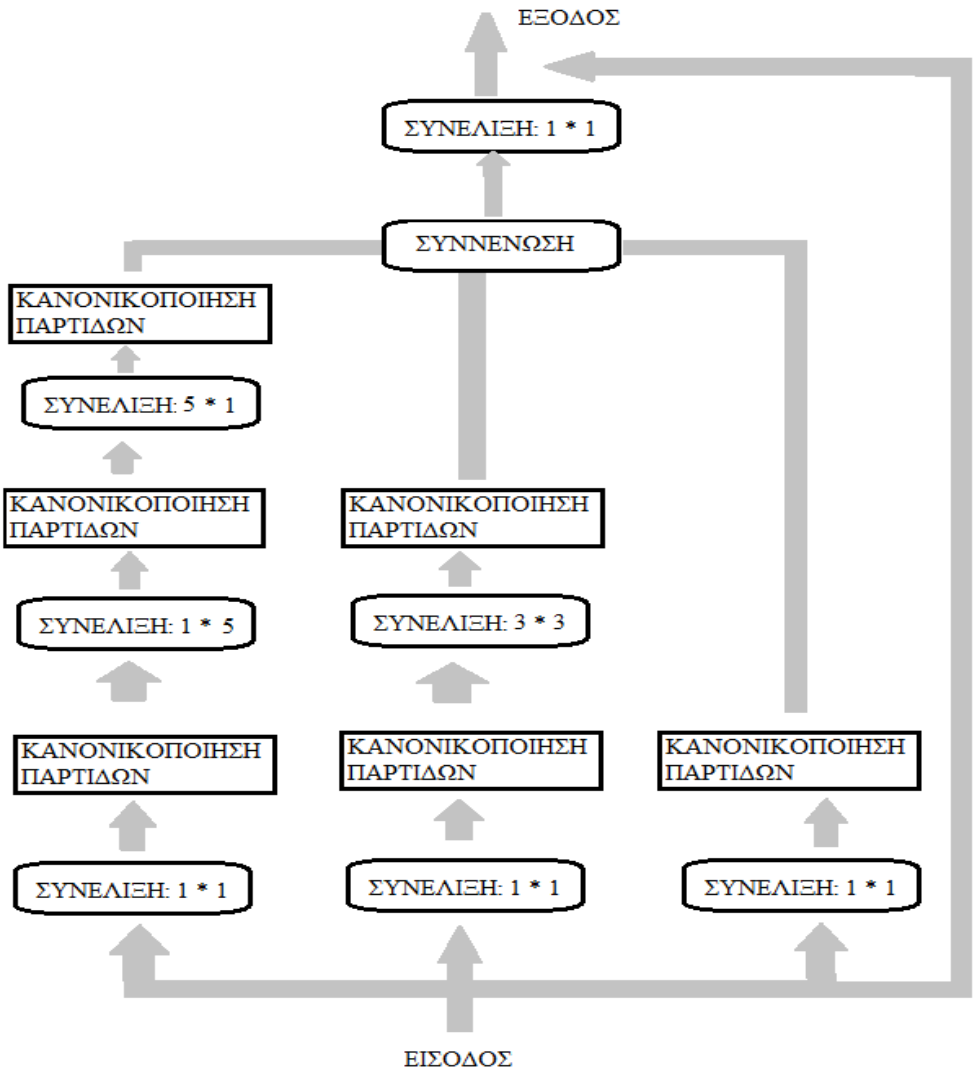
Inception-SSD

Η αρχιτεκτονική του μοντέλου Inception-SSD είναι παρόμοια με αυτήν του παραπάνω MobileNet-SSD. Η διαφορά είναι ότι η βασική αρχιτεκτονική εδώ είναι το μοντέλο Inception. Το επιπλέον επίπεδο αλγορίθμου SSD περιέχει περιορισμένες πληροφορίες μικρού στόχου και έχει μόνο $3 * 3$ πυρήνα συνέλιξης, τότε υπάρχει ένα πρόβλημα με τις ελλείψεις λεπτομερειών στόχου. Ταυτόχρονα, το SSD δίκτυο αλγορίθμου έχει μια σύνθετη δομή μοντέλου νευρωνικού δικτύου. Για τη βελτίωση της απόδοσης του δικτύου, η γενική μέθοδος είναι η αύξηση του πλάτους και του βάθους του δικτύου. Ωστόσο, θα οδηγήσει σε αύξηση του αριθμού των παραμέτρων δικτύου και με αποτέλεσμα αύξηση του

ποσού των υπολογισμών και το πρόβλημα της υπερβολικής προσαρμογής. Επομένως, προτείνεται το δίκτυο Inception-SSD να χρησιμοποιηθεί, που εισάγεται στο δίκτυο SSD, και η ακρίβεια ανίχνευσης του δικτύου βελτιώνεται σημαντικά.

Η αρχιτεκτονική Inception έχει την υψηλή απόδοση του πυκνού ή σποραδικού πίνακα και διατηρεί την αραιή δομή του δικτύου, η οποία μπορεί να λύσει το πρόβλημα της υπερβολικής προσαρμογής και του αυξημένου υπολογισμού κατά τη διάρκεια της βελτιστοποίησης των νευρωνικών δικτύων.

Προκειμένου να διατηρηθούν οι αρχικές λεπτομέρειες του στόχου, εισάγεται το μπλοκ Inception για να αντικαταστήσει μερικά επιπλέον επίπεδα στο αρχικό δίκτυο SSD και το μπλοκ Inception έχει βελτιστοποιηθεί για την αποσύνθεση των $5 * 5$ πυρήνα συνελεύσεων σε δύο μονοδιάστατες συνελεύσεις ($1 * 5$ και $5 * 1$). Έτσι χωρίζοντας 1 μεταβολή σε 2 μετατρ., εμβαθύνει περαιτέρω το δίκτυο και αυξάνει τη μη γραμμικότητα του δικτύου εντοπισμού στόχου. Η πλεονάζουσα υπολογιστική ισχύς χρησιμοποιείται για την εμβάθυνση του δικτύου, ταυτόχρονα, ο αριθμός των χαρακτηριστικών των χάρτων σε κάθε επίπεδο του δικτύου αντίληψης μειώνεται, έτσι ώστε το άθροισμα των χαρτών χαρακτηριστικών να είναι το ίδιο με αυτό του αρχικού δικτύου SSD. Προκειμένου να αντανακλά τη σημασία των διαφόρων κλιμάκων του δεκτικού πεδίου, οι τρεις πυρήνες συνελεύσεων ($1 * 1, 3 * 3, 5 * 5$) σταθμίζονται ξεχωριστά και σταθμίζονται με τιμή $w = \{1, 2, 1\}$. Στην ενότητα Inception, κάθε στρώμα συνέλιξης τόμου ακολουθείται από Κανονικοποίηση Παρτίδων (Batch Normalization, BN). Επιπλέον, προσθέτουμε το επίπεδο συνέλιξης $\mathrm{Conv}\ 1 * 1$ κοντά στην έξοδο για να μειώσουμε τον αριθμό παραμέτρων και να βελτιώσουμε τη ταχύτητα υπολογισμού.



Εικόνα 26: Το αρχικό δομικό στοιχείο με BN για την δομή Inception-SSD.

Το μοντέλο που βασίζεται στη δομή Inception-SSD φαίνεται στην εικόνα 26. Το Inception-SSD υιοθετεί την πλήρη μέθοδο συνέλιξης, αν και το δεκτικό πεδίο είναι ολόκληρη η εικόνα, πρέπει ακόμη να ταξινομηθεί η καθεμία με δεκτικό πεδίο για διαφορετικές κλίμακες και σχήματα. Το Inception-SSD θα προβλέψει άμεσα την κατηγορία του και πληροφορίες τοποθεσίας με παλινδρόμηση όπου πραγματοποιείται με διασπορά σε διαφορετικά επίπεδα. Έτσι βελτιώνεται η συνολική ακρίβεια και ταχύτητα του δικτύου και αξιοποιούμε πλήρως το πλεονέκτημα της πυκνής δομής μήτρας. Στη συνέχεια, βελτιώνουμε την ακρίβεια ανίχνευσης μικρών στόχων όταν βελτιστοποιείται το νευρικό δίκτυο.

ΚΕΦΑΛΑΙΟ 3

Εφαρμογές μοντέλων ανίχνευσης αντικειμένων και αποτελέσματα

Εργαλεία και Βιβλιοθήκες Υλοποίησης στη δική μας εφαρμογή
Tensorflow & Keras

Το πιο γνωστό εργαλείο που χρησιμοποιείται για την ανάπτυξη νευρωνικών δικτύων είναι το tensorflow. Το Tensorflow είναι μια πλατφόρμα ανοιχτού κώδικα για μηχανική μάθηση. Παρέχει ένα ολοκληρωμένο οικοσύστημα εργαλείων, για προγραμματιστές, επιχειρήσεις και ερευνητές που θέλουν να προωθήσουν τις τελευταίες τέχνες της μηχανικής μάθησης και να δημιουργήσουν επεκτάσιμες εφαρμογές μηχανικής μάθησης. Το tensorflow έχει σχεδιαστεί για να μας βοηθήσει να μάθουμε να δημιουργούμε μοντέλα εύκολα, με ένα διαισθητικό εύχρηστο σύνολο API που μας διευκολύνει να μάθουμε και να εφαρμόζουμε μηχανική εκμάθηση, βαθιά μάθηση και επιστημονική πληροφορική. Μας παρέχει μια πλούσια συλλογή εργαλείων για κατασκευή μοντέλων. Αυτές περιλαμβάνουν προεπεξεργασία δεδομένων, απορρόφηση δεδομένων, οπτικοποίηση αξιολόγησης μοντέλου και προβολή, αλλά δεν είναι μόνο για δημιουργία μοντέλων. Δημιουργήθηκε και εφαρμόστηκε το 2011 από την Google. Διατίθεται σε πολλές εκδώσεις και μπορεί να χρησιμοποιηθεί με CPU ή GPU (με εφαρμογή της βιβλιοθήκης CUDA). Είναι γνωστό ότι η εκτέλεση νευρωνικών δικτύων είναι προτιμότερη στη κάρτα γραφικών, πετυχαίνοντας πολύ γρηγορότερη εκπαίδευση του μοντέλου. Μπορούμε εύκολα να εκπαιδεύσουμε και να αναπτύξουμε το μοντέλο μας οπουδήποτε με tensorflow.

Εικόνα 27: Λογότυπο του Tensorflow.

(Πηγή: https://favpng.com/png_view/google-tensorflow-google-brain-machine-learning-deep-learning-png/gkt8s73w)

Το μοντέλο εκπαιδεύτηκε σε υπολογιστή με τα παρακάτω χαρακτηριστικά

- **GPU:** GTX 1060
- **CPU:** Intel i7-4500
- **RAM:** 12 GB
- **Disk:** 250 GB

Για την εφαρμογή του μοντέλου και την δημιουργία των απαιτούμενων νευρωνικών δικτύων χρειάστηκε να γίνει χρήση της βιβλιοθήκης Keras. Αυτό έγινε διότι με την επιλογή της βιβλιοθήκης Keras παρακάμπτουμε τον πιο αργό προγραμματισμό του μοντέλου από το Tensorflow. Το Keras, το επιτυγχάνει με χρήση πιο απλών μεθόδων και εντολών για την υλοποίηση των μοντέλων.

Εικόνα 28: Λογότυπο του Keras.

(Πηγή: https://keras.io/)

Python & OpenCV

Για την υλοποίηση της διπλωματικής εργασίας χρησιμοποιήθηκαν πολλά module της Python, όπως επίσης χρησιμοποιήθηκε και η numpy.

Επίσης, έγινε χρήση OpenCV βιβλιοθήκης, για τις χρήσιμες συναρτήσεις της, όπου μας βοηθά για μεταχείριση εικόνων κ.α.

Εικόνα 29: Λογότυπο του OrenCV και Python.

(Πηγή: https://opencv.org/opencv-python-is-now-an-official-opencv-project/)

Επιβλεπόμενη μάθηση

Χρησιμοποιήσαμε την τεχνική της επιβλεπόμενης μάθησης, ώστε τα νευρωνικά δίκτυα να εκπαιδευτούν. Μια κατηγορία της μηχανικής μάθησης είναι και η επιβλεπόμενη μάθηση. Σκοπός της είναι να αντλήσει χαρακτηρισμούς από τα δεδομένα εκπαίδευσης ώστε να μπορεί να εντοπίζει αυτούς τους χαρακτηρισμούς σε πιο γενικά δεδομένα.

Για να πετύχει αυτό, τα δεδομένα εκπαίδευσης απαρτίζονται από ένα άθροισμα παραδειγμάτων από τα χαρακτηριστικά που μας ενδιαφέρουν για να γίνει η εκπαίδευση των μοντέλων.
Κάθε φορά που εισέρχονται στην είσοδο δεδομένα αντιστοιχούν σε επιθυμητές τιμές εξόδου. Η δουλειά των αλγόριθμων επιβλεπόμενης μάθησης είναι να επεξεργάζονται τα δεδομένα εκπαίδευσης και να δημιουργούν ένα μοντέλο που μπορεί να αξιοποιηθεί για να κατηγοριοποιήσει νέα παραδείγματα. Ο στόχος του αλγορίθμου είναι να επιτύχει τη σωστή κατηγοριοποίηση για τα άγνωστα δεδομένα που εισέρχονται ως παραδείγματα. Για να έχουμε καλύτερα αποτελέσματα στη κατηγοριοποίηση από άγνωστα σύνολα δεδομένων, πρέπει να γίνει γενίκευση των δεδομένων εκπαίδευσης από τον αλγόριθμο μάθησης.

Το σύνολο δεδομένων που χρησιμοποιήθηκε, δημιουργήθηκε για την παρούσα διπλωματική εργασία και διαιρέθηκε σε δύο μέρη. Σε 20% για το validation set και 80% για το train set, αντίστοιχα. Οι ετικετές ήταν οι εξής: Σφαίρα και Σπιρτόκουτο.

Εκπαίδευση Νευρωνικών Δικτύων στο δικό μας dataset

Στην ενότητα αυτή θα παρουσιάσουμε τα αποτελέσματα της εκπαίδευσης των νευρωνικών δικτύων από το δικό μας dataset, που εκπαιδεύτηκαν με χρήση CPU και GPU. Έπειτα, θα περιγράψουμε τη διαδικασία που πραγματοποιήσαμε για την εκπαίδευση και θα αναφέρουμε τα δεδομένα των εφαρμογών μας.

Επιλογή του dataset.

Η επιλογή του dataset έγινε με σκοπό να προσπεράσουμε κάποια τεχνικά προβλήματα. Αν και γνωρίζουμε ότι η εκπαίδευση των αλγορίθμων νευρωνικών δικτύων έχει καλύτερα αποτελέσματα με ένα dataset με όσα περισσότερα δεδομένα γίνεται. Λόγω της περιορισμένης

ισχύς του ηλεκτρονικού υπολογιστή που είχαμε στη κατοχή μας αλλά και για τη μείωση του χρόνου εκπαίδευσης των πειραμάτων μας, επιλέξαμε να δημιουργήσουμε ένα δικό μας dataset μειωμένο σε μέγεθος, τέτοιο ώστε να επιτευχθεί η ανίχνευση αντικειμένων αλλά προφανώς με μικρότερη ακρίβεια. Η έκβαση της ακρίβειας δε μας απασχολεί στα δικά μας πειραματικά αποτελέσματα σε αυτή την ενότητα, μας απασχολεί όμως η σύγκριση της ταχύτητας ανίχνευσης αντικειμένων CPU με GPU.

Στο dataset που δημιουργήσαμε προσέξαμε να κάνουμε ποσοστιαία ισοκατανομή των δειγμάτων στις κλάσεις: Σφαίρα και Σπιρτόκουτο.

Το TensorFlow χρειάζεται εκατοντάδες εικόνες ενός αντικειμένου για να εκπαιδεύσει έναν καλό ταξινομητή ανίχνευσης. Για να εκπαιδεύσουμε έναν ισχυρό ταξινομητή, οι εικόνες εκπαίδευσης θα πρέπει να έχουν τυχαία αντικείμενα στην εικόνα μαζί με τα επιθυμητά αντικείμενα και θα πρέπει να έχουν ποικιλία φόντων και συνθήκες φωτισμού. Πρέπει να υπάρχουν κάποιες εικόνες όπου το επιθυμητό αντικείμενο είναι μερικώς σκοτεινό, επικαλυπτόμενο με κάτι άλλο ή μόνο στη μέση της εικόνας.

Το σύνολο δεδομένων αποτελείται από 311 εικόνες με δυο πεδία ανίχνευσης (Σφαίρα και Σπιρτόκουτο), η ανάλυση των εικόνων δεν υπερβαίνει τα 720x1280 και τα 200KB και έγινε χρήση του script resizer.py για να μειώσουμε το μέγεθος των εικόνων.

Περιγραφή αλγορίθμου εκπαίδευσης

Σε αυτή την ενότητα θα περιγράψουμε τη διαδικασία εκπαίδευσης που πραγματοποιήσαμε. Τα βήματα του πρώτου σταδίου είναι τα ακόλουθα:

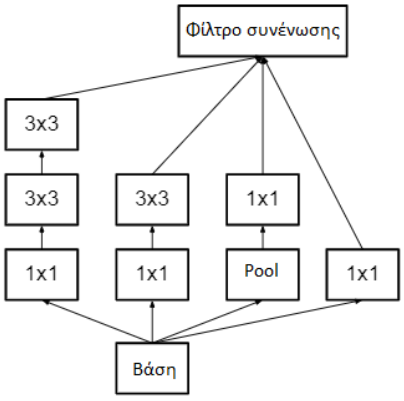
- **Συγκέντρωση δεδομένων:** Συλλέχτηκαν 311 εικόνες.
- **Ομαδοποίηση - Επισήμανση αντικείμενων:** Με όλες τις εικόνες που συγκεντρώθηκαν, επισημαίνουμε τα επιθυμητά αντικείμενα σε κάθε εικόνα. Αυτό έγινε με την βοήθεια του εργαλείου LabelImg, Το LabelImg αποθηκεύει ένα αρχείο .xml που περιέχει τα δεδομένα ετικέτας για κάθε εικόνα. Αυτά τα αρχεία .xml θα χρησιμοποιηθούν για τη δημιουργία TFRecords, τα οποία είναι μία από τις εισόδους στον εκπαιδευτή TensorFlow. Μόλις επισημάνουμε και αποθηκεύσουμε κάθε εικόνα, θα υπάρχει ένα αρχείο .xml για κάθε εικόνα στους καταλόγους.

- **Διαχωρισμός Dataset:** Μετακινήθηκε το 20% των εικόνων – δεδομένων στον κατάλογο test και το 80% στον κατάλογο train. Βεβαιωθήκαμε ότι υπάρχει μια ποικιλία εικόνων στους καταλόγους test και train.
- **Κατασκευή Δικτύων:** Η κατασκευή των δικτύων CNN πραγματοποιήθηκε με χρήση των εργαλείων Tensorflow και Keras. Με τη σωστή επιλογή επιπέδων και τον κατάλληλο αριθμό νευρώνων.
- **Testing Μοντέλου:** Έγινε η αξιολόγηση του μοντέλου σε άγνωστα δεδομένα. Η απόδοση του μοντέλου στο συγκεκριμένο σημείο είναι η πιο σημαντική διότι από εδώ μπορούμε να καταλάβουμε πόσο καλά έχει εκπαιδευτεί το μοντέλο.

Εφαρμογή και αποτελέσματα με χρήση CPU-GPU

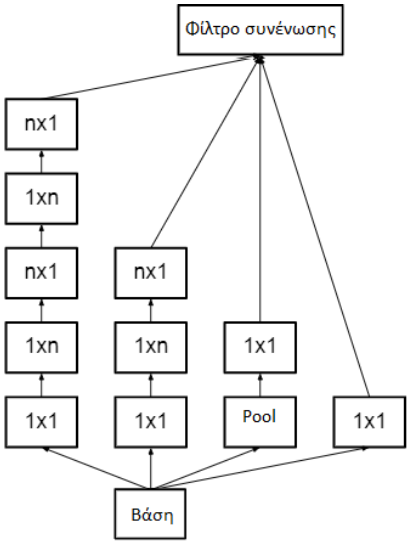
Σε αυτή την ενότητα γίνεται αναφορά για την εφαρμογή του μοντέλου Faster-R-CNN-Inception-V2, που έγινε με χρήση της CPU και GPU χρησιμοποιώντας τις ίδιες παραμετροποιήσεις σε διαφορά μεγέθη ανάλυσης της dataset με σκοπό να μπορέσουμε να εξάγουμε διάφορα αποτελέσματα. Δεν θα δώσουμε τόσο έμφαση στην ακρίβεια όσο στους χρόνους που κάνει το μοντέλο για κάθε φάση του dataset στην εφαρμογή του, όσο στη χρήση CPU και GPU αντιστοίχως.

Στην εφαρμογή μας γίνεται χρήση των νευρωνικών δικτύων Inception V2. Στην αρχιτεκτονική του, το επίπεδο συνέλιξης $5 * 5$ της Inception V1 αντικαθίσταται με δύο επίπεδα συνέλιξης $3 * 3$. Έτσι, η χρήση δύο επιπέδων $3 * 3$ αντί για $5 * 5$ αυξάνει την απόδοση της αρχιτεκτονικής.



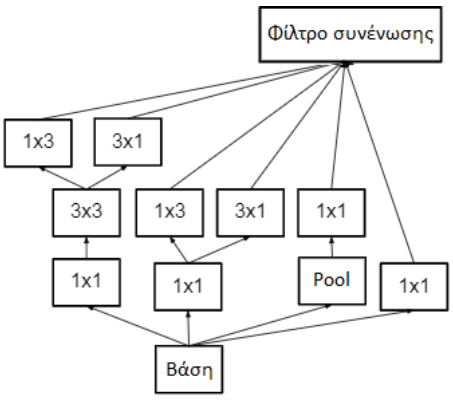
Εικόνα 30: Δομική μορφή της Inception V2.

Αυτή η αρχιτεκτονική μετατρέπει επίσης την παραγοντοποίηση $n * n$ σε παραγοντοποίηση $1 * n$ και $n * 1$. Επομένως, μια συνέλιξη $3 * 3$ μπορεί να μετατραπεί σε $1 * 3$ και στη συνέχεια ακολουθείται από $3 * 1$, η οποία είναι 33% φθηνότερη όσον αφορά την υπολογιστική πολυπλοκότητα σε σύγκριση με το $3 * 3$.



Εικόνα 31

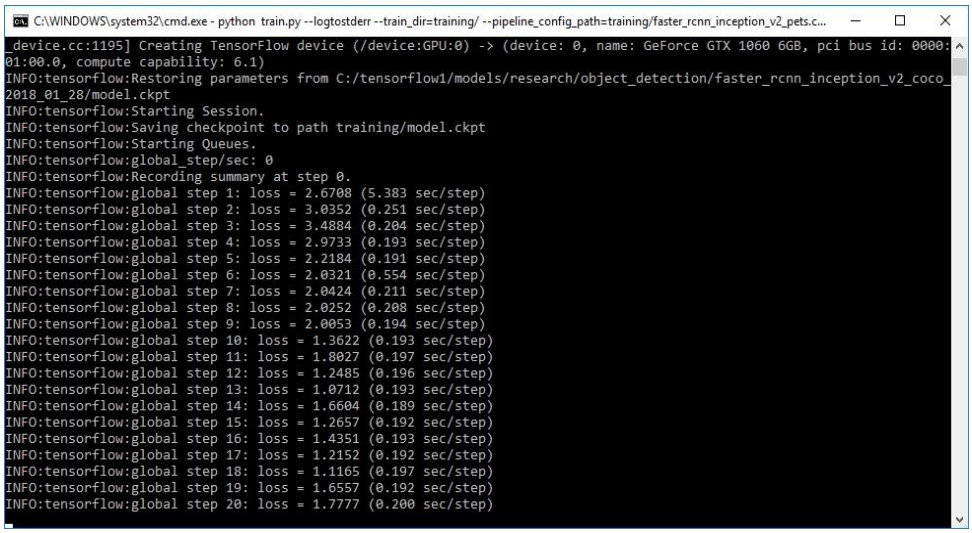
Για να αντιμετωπιστεί το πρόβλημα της αντιπροσωπευτικής συμφόρησης, οι τράπεζες χαρακτηριστικών της μονάδας επεκτάθηκαν αντί να την κάνουν πιο βαθιά. Αυτό θα αποτρέψει την απώλεια πληροφοριών που προκαλεί όταν τις κάνουμε βαθύτερες.



Εικόνα 32

Έγιναν συνολικά οχτώ εφαρμογές του μοντέλου Faster-R-CNN-Inception-V2. Μια για το 100% της ανάλυσης των εικόνων του dataset, έπειτα για το 75%, μετά για το 50% και τέλος για το 25% με χρήση CPU. Έγινε επανάληψη της διαδικασίας και με χρήση της GPU.

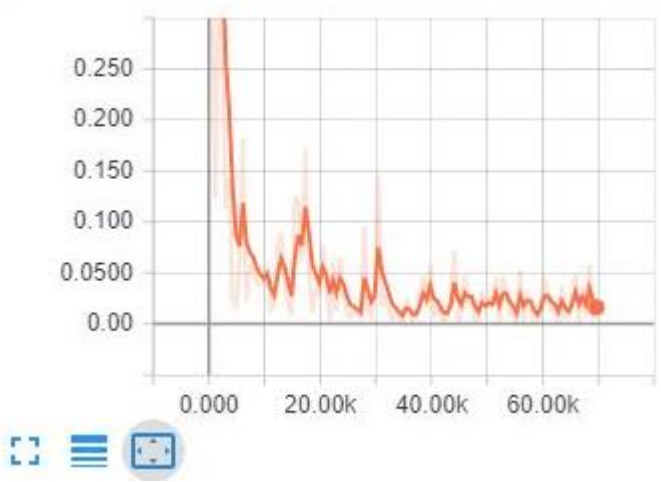
Για χάριν συντομίας παρουσιάζεται μια σειρά από εικόνες που αφορούν το 75% του dataset με χρήση GPU.



Εικόνα 33: Στιγμιότυπο από το cmd των Windows κατά τη διάρκεια εκπαίδευσης.

Στην εικόνα παρουσιάζεται το στάδιο της εκπαίδευσης, μπορούμε να διακρίνουμε τη συσκευή που τρέχει (CPU-GPU) και τα βήματα (steps) της εκπαίδευσης.

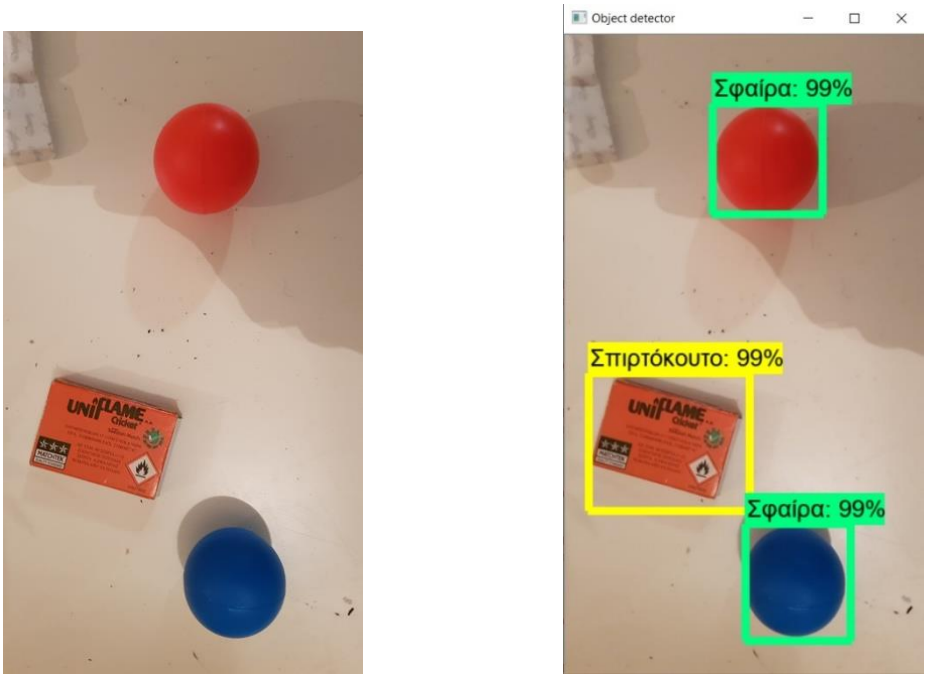
Κάθε βήμα της εκπαίδευσης αναφέρει την απώλεια. Θα ξεκινήσει με υψηλή απώλεια και θα κατεβαίνει όλο και πιο χαμηλά καθώς προχωρά η εκπαίδευση. Για την εκπαίδευσή της φωτογραφίας, η απώλεια ξεκίνησε περίπου στο 3,0 και έπεσε γρήγορα κάτω από το 0,8. Έπειτα το μοντέλο προπονήθηκε έως ότου η απώλεια μειωθεί σταθερά κάτω από 0,05, κάτι που θα διαρκέσει περίπου 40.000 βήματα ή περίπου 2 ώρες (αυτό εξαρτάται από το πόσο ισχυρή είναι η CPU και η GPU).



Εικόνα 34: Στιγμιότυπο από το TensorBoard κατά τη διάρκεια εκπαίδευσης του δικού μας dataset.

Στην παραπάνω εικόνα παρουσιάζεται η απώλεια για την ταξινόμηση των αντικειμένων που εντοπίστηκαν στις κατηγορίες: Σφαίρα και Σπιρτόκουτο, συναρτήσει των βημάτων. Παρατηρούμε ότι στις πρώτες 10.00k επαναλήψεις βημάτων εντοπίζεται η μεγαλύτερη πτώση απώλειας στο γράφημα.

Μετά την ολοκλήρωση της εκπαίδευσης, κάναμε εφαρμογή από ένα πακέτο φωτογραφιών που συμπεριλαμβάνουν τα αντικείμενα ανίχνευσής μας για να δούμε τη λειτουργικότητα του μοντέλου μας από την εκπαίδευση. Συγκεκριμένα, οι ακόλουθες φωτογραφίες δε συμπεριλαμβάνονται στο άνωθεν πακέτο, αλλά στο πακέτο εκπαίδευσης και για αυτό παρατηρούμε μεγάλη ακρίβεια στην ανίχνευση των αντικειμένων μας. Το πακέτο που χρησιμοποιήθηκε για την ανίχνευση της ακρίβειας δεν είχε χρησιμοποιηθεί για την εκπαίδευση και είναι άγνωστο για το μοντέλο μας, εκεί δεν επιτεύχθηκε τόσο μεγάλη ακρίβεια.



Εικόνα 35: Εμφάνιση του αποτελέσματος της εφαρμογής μας από τα αντικείμενα ανίχνευσής μας.

Η καλύτερη ακρίβεια καταγράφηκε στο μοντέλο που έγινε εκπαίδευση με full size image του dataset και ήταν 83.40%. Αντίστοιχα με 75% το size image του dataset επιτεύχθηκε 80% ακρίβεια και ακολούθησε το 72.20% και το 53.60% αντιστοίχως για το 50% και το 25%. Αυτή η διαφορά ακρίβειας από την Εικόνα 35 οφείλεται από το μικρό όγκο του dataset που δημιουργήσαμε, ώστε να μπορέσουμε να ξεπεράσουμε τα προβλήματα του hardware ως προς το χρόνο και την ισχύ του. Στη συγκεκριμένη εφαρμογή, μας ενδιαφέρει κυρίως να μελετήσουμε το χρόνο απόκρισης και όχι την ακρίβεια.

Έτσι λοιπόν, παραθέτουμε κάτω τον πίνακα των αποτελεσμάτων από τις εφαρμογές του μοντέλου μας για τις περιπτώσεις 100%, 75%, 50% και 25% με χρήση της CPU i7 και της GTX 1060 αντιστοίχως.

	Μέγεθος εικόνας (100%)	Μέγεθος εικόνας (75%)	Μέγεθος εικόνας (50%)	Μέγεθος εικόνας (25%)
Avg. mAP (%)	83.40	80.00	72.20	53.60
CPU i7 (sec)	6.220	3.444	1.545	0.405
GTX 1060 (sec)	0.371	0.233	0.122	0.053
Impv GPU vs CPU (%)	94.03	93.23	92.10	86.91

Πίνακας 1: Αποτελέσματα της εφαρμογής.

Το Impv GPU vs CPU (%) δηλώνει ποσοστιαία πόσο ταχύτερη είναι η υλοποίηση με τη χρήση της GPU από τη CPU.

Βιβλιογραφικά δεδομένα

Σύμφωνα με το επιστημονικό άρθρο: «Appliacation of deep learning in object detection» που επιμελήθηκαν οι συγγραφείς: Xinyi Zhou, Wei Gong, WenLong Fu και Fengtong Du, συγκρίνανε το μέσο όρο ακρίβειας (mAP) με διάφορα είδη δομής δικτύου στο σύνολο δεδομένων VOC2007 και καταλήξανε στα εξής αποτελέσματα:

Network	R-CNN	Fast R-CNN	Faster R-CNN
VOC07 mAP	0.66	0.669	0.732

Πίνακας 2: [15].

Από τις βιβλιογραφικές αναφορές μας [8], [12], [13] και [14] αντλούνται τα ακόλουθα δεδομένα του πίνακα 3.

Μέθοδος	Χρόνος εκπαίδευσης (h)	Χρόνος δοκιμής (sec)
R-CNN	84	49
Fast R-CNN	8.75	2.3
Faster R-CNN	-	0.2

Πίνακας 3: Σύγκριση ταχύτητας χρόνου εκπαίδευσης και χρόνου δοκιμής.

Ο λόγος για τον οποίο το "Fast R-CNN" είναι γρηγορότερο από το R-CNN είναι επειδή δεν χρειάζεται να τροφοδοτείται 2000 προτάσεις περιοχής στο συνελικτικό νευρωνικό δίκτυο κάθε φορά. Αντί αυτού, η λειτουργία συνέλιξης γίνεται μόνο μία φορά ανά εικόνα και δημιουργείται ένας χάρτης χαρακτηριστικών από αυτήν.

Από το πίνακα 3, μπορούμε να συμπεράνουμε ότι το Fast R-CNN είναι σημαντικά ταχύτερο σε προπονήσεις και δοκιμές σε σχέση με το R-CNN. Όταν εξετάζουμε την απόδοση του Fast R-CNN κατά τη διάρκεια του χρόνου δοκιμής, συμπεριλαμβανομένων των προτάσεων περιοχής επιβραδύνεται σημαντικά ο αλγόριθμος σε σύγκριση με τη μη χρήση προτάσεων περιοχής. Επομένως, οι προτάσεις περιοχής γίνονται εμπόδια στον αλγόριθμο Fast R-CNN που επηρεάζει την απόδοσή του. Ακόμη, διαπιστώνουμε ότι το Faster R-CNN είναι πολύ πιο γρήγορο από τους προκατόχους του. Επομένως, μπορεί ακόμη και να χρησιμοποιηθεί για την ανίχνευση αντικειμένων σε πραγματικό χρόνο.

Από τη βιβλιογραφική αναφορά [6], έχουμε κάποια αποτελέσματα για το σύνολο δοκιμών PASCAL VOC 2007 που φαίνονται στο παρακάτω πίνακα.

Μέθοδος	Δεδομένα εκπαίδευσης	mAP (%)	FPS
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ResNet	2007+2012	76.4	5
YOLO	2007+2012	63.4	45

Πίνακας 4: Πλαίσια ανίχνευσης στο PASCAL VOC 2007.

Από το πίνακα 4 διαπιστώνουμε ότι η απόδοση ακρίβειας του Faster R-CNN είναι καλύτερη από τους προκατόχους τους και από το YOLO. Το YOLO όμως, έχει μεγάλο πλεονέκτημα στη ταχύτητα.

Μέθοδος	Top – 1 ακρίβειας	Πλήθος παραμέτρων
VGG-16	71.0	14,714,688
MobileNet	71.1	3,191,072
Inception V2	73.9	10,173,112

Πίκανας 5: Ιδιότητες 3 εξολκέων χαρακτηριστικών.

Η ακρίβεια Top-1 είναι η ακρίβεια ταξινόμησης στο ImageNet και στο πίνακα 5 αντλούμενος από τη βιβλιογραφική αναφορά [5], παρατηρούμε ότι το Inception V2 έχει τη μεγαλύτερη ακρίβεια. Όσον αφορά το πλήθος των παραμέτρων το VGG-16 έχει το μεγαλύτερο πλήθος (14,714,688) και τη το MobileNet έχει το μικρότερο πλήθος (3,191,072).

Κεφάλαιο 4

Συμπεράσματα

Η ανίχνευση αντικειμένων είναι ένα συναρπαστικό πεδίο και ορθώς βλέπει έναν τόνο έλξης τόσο σε εμπορικές όσο και σε ερευνητικές εφαρμογές. Χάρη στην πρόοδο του σύγχρονου υλικού και των υπολογιστικών πόρων, οι καινοτομίες σε αυτόν τον χώρο είναι γρήγορες και πρωτοποριακές.

Οι μέθοδοι R-CNN ήταν πραγματικά ένα παιχνίδι αλλαγής για εργασίες εντοπισμού αντικειμένων. Υπήρξε ξαφνικά μια αύξηση στα τελευταία χρόνια στον αριθμό των εφαρμογών όρασης υπολογιστών που δημιουργούνται και το R-CNN βρίσκεται στην καρδιά των περισσότερων από αυτές. Στο παρακάτω πίνακα είναι μια περίληψη των R-CNN μεθόδων.

Μέθοδος	Χαρακτηριστικά	Χρόνος / εικόνα πρόβλεψης	Περιορισμοί
CNN	Χωρίζει την εικόνα σε πολλές περιοχές και στη συνέχεια, ταξινομεί κάθε περιοχή σε διάφορες κατηγορίες.	-	Χρειάζεται πολλές περιοχές για να προβλέψει με ακρίβεια και ως εκ τούτου υψηλό χρόνο υπολογισμού.
R-CNN	Χρησιμοποιεί επιλεκτική αναζήτηση για τη δημιουργία περιοχών. Εξάγει περίπου 2000 περιοχές από κάθε εικόνα.	40-50 δευτερόλεπτα	Υψηλός χρόνος υπολογισμού καθώς κάθε περιοχή περνά στο CNN ξεχωριστά, επίσης χρησιμοποιεί τρία διαφορετικά μοντέλα για να κάνει προβλέψεις.
Fast R-CNN	Κάθε εικόνα μεταφέρεται μόνο μία φορά στο CNN	2 δευτερόλεπτα	Η επιλεκτική αναζήτηση είναι αργή και ως εκ

	και εξάγονται χάρτες χαρακτηριστικών. Επιλεκτική αναζήτηση χρησιμοποιείται σε αυτούς τους χάρτες για τη δημιουργία προβλέψεων. Συνδυάζει και τα τρία μοντέλα που χρησιμοποιούνται στο R-CNN μαζί.		τούτου ο χρόνος υπολογισμού εξακολουθεί να είναι υψηλός.
Faster R-CNN	Αντικαθιστά την επιλεκτική μέθοδο αναζήτησης με δίκτυο πρότασης περιοχής που έκανε την μέθοδο πολύ πιο γρήγορο.	0.2 δευτερόλεπτα	Η πρόταση αντικειμένου απαιτεί χρόνο και καθώς υπάρχουν διαφορετικά συστήματα που λειτουργούν το ένα μετά το άλλο, η απόδοση των συστημάτων εξαρτάται από την απόδοση του προηγούμενου συστήματος.

Πίνακας 6: Συγκεντρωτικός πίνακας των R-CNN μεθόδων

Με βεβαιότητα μπορούμε να πούμε ότι το Faster R-CNN είναι πιο αποτελεσματικό από την οικογένεια των R-CNN μεθόδων.

Επίσης, οι δύο μέθοδοι YOLO και Faster R-CNN έχουν κάποιες ομοιότητες. Χρησιμοποιούν μια δομή δικτύου βασισμένη σε κουτιά αγκύρωσης και χρησιμοποιούν οριοθέτηση της παλινδρόμησης. Οι διαφορές του YOLO από Faster R-CNN είναι ότι κάνει την

ταξινόμηση και την οπισθοδρόμηση του πλαισίου οριοθέτησης ταυτόχρονα. Είναι λογικό ότι η YOLO σαν μεταγενέστερη έχει έναν πιο κομψό τρόπο να κάνει παλινδρόμηση και ταξινόμηση.

Ωστόσο, το YOLO έχει το μειονέκτημά του στην ανίχνευση αντικειμένων. Δυσκολεύεται να εντοπίσει αντικείμενα που είναι μικρά και κοντά το ένα στο άλλο λόγω μόνο δύο κουτιών αγκύρωσης σε ένα πλέγμα που προβλέπει μόνο μία κατηγορία αντικειμένων. Δεν γενικεύεται καλά όταν τα αντικείμενα στην εικόνα εμφανίζουν σπάνιες πτυχές αναλογίας. Το Faster R-CNN από την άλλη πλευρά, ανιχνεύει μικρά αντικείμενα καλά επειδή έχει εννέα κουτιά αγκύρωσης σε ένα μόνο πλέγμα, ωστόσο αποτυγχάνει να κάνει ανίχνευση σε πραγματικό χρόνο με την αρχιτεκτονική των δύο βημάτων.

Το FPN είναι χρονοβόρο και απαιτεί υψηλή μνήμη για να εκπαιδευτεί ταυτόχρονα από άκρο σε άκρο. Ακόμη, μπορούμε να το αξιοποιήσουμε μόνο για την υψηλή ακρίβειά του κι έχει καλά αποτελέσματα σε ανίχνευση μικρών αντικειμένων.

Το Mobilenet θεωρείται ελαφρύ πρόγραμμα, έτσι, φορτώνει γρήγορα και κάνει προβλέψεις γρήγορες αλλά όχι τόσο ακριβείς. Έχει ευρεία εφαρμογή σε εφαρμογές κινητών.

Επιπρόσθετα παρατηρούμε ότι τα μεγάλα dataset είναι η βάση της επιτυχίας της βαθιάς μάθησης, είναι τα καύσιμα για τον πύραυλο της βαθιάς μάθησης. Ωστόσο, η ποιότητα του dataset επηρεάζει τη βαθιά μάθηση στη πράξη. Από τα δεδομένα που παρουσιάστηκαν στα προηγούμενα κεφάλαια, φαίνεται ξεκάθαρα η επίτευξη καλύτερης ακρίβειας σε αλγόριθμους που έχουν εκπαιδευτεί με μεγαλύτερα dataset σε αντίθεση με τα μικρότερα dataset.

Όσον αφορά την ανάλυση εικόνας εισόδου μπορούμε να πούμε ότι η υψηλότερη ανάλυση βελτιώνει σημαντικά την ανίχνευση αντικειμένων για μικρά αντικείμενα, ενώ βοηθά επίσης σε μεγάλα αντικείμενα. Όταν μειώνεται η ανάλυση κατά συντελεστή δύο και στις δύο διαστάσεις, η ακρίβεια μειώνεται κατά 15,88% κατά μέσο όρο, αλλά ο χρόνος συμπερασμάτων μειώνεται επίσης κατά έναν συντελεστή 27,4% κατά μέσο όρο. Ωστόσο, με την ανίχνευση μίας σταθερής εικόνας, κερδίζουμε ταχύτητα με κόστος ακρίβειας.

Τέλος, σε ερώτημα ποιο μοντέλο ανίχνευσης αντικειμένων μπορούμε να επιλέξουμε, αυτό εξαρτάται από τη συγκεκριμένη απαίτησή μας. Το πιο σημαντικό ερώτημα δεν είναι ποιος ανιχνευτής είναι ο καλύτερος. Το πραγματικό ερώτημα είναι ποιος ανιχνευτής και ποιες διαμορφώσεις μας δίνουν την καλύτερη ισορροπία ταχύτητας και ακρίβειας που απαιτείται από την εφαρμογή μας.

Με το Faster R-CNN, θα έχουμε υψηλή ακρίβεια αλλά αργή ταχύτητα. Εάν θέλουμε ένα μοντέλο υψηλής ταχύτητας που μπορεί να λειτουργήσει στον εντοπισμό ροής βίντεο σε υψηλό fps, το δίκτυο ανίχνευσης μίας λήψης YOLO λειτουργεί καλύτερα σχετικά με τις μεθόδους που έχουμε προαναφέρει.

Βιβλιογραφία

[1] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe and Jonathon Shlens, ¨Rethinking the Inception Architecture for Computer Vision¨, arXiv:1512.00567v3, London, 2015.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, ¨Going deeper with convolutions¨, arXiv:1409.4842v1, 2014.

[3] Ioffe, S., & Szegedy, C., ¨Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift¨, arXiv:1502.03167v3, 2015.

[4] J. Bao, D. Chen, F. Wen, H. Li and G. Hua, ¨Towards open-set identity preserving face synthesis¨, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), arXiv:1803.11182v2, 2018.

[5] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama and Kevin Murphy, ¨Speed / accuracy trade-offs for modern convolutional object detectors¨, arXiv:1611.10012v3, 2017.

[6] Joseph Redmon and Ali Farhadi, ¨YOLO9000: Better, Faster, Stronger¨, arXiv:1612.08242v1, 2016.

[7] Joseph Redmon and Ali Farhadi, ¨Yolov3: An incremental improvement¨, arXiv:1804.02767v1, Washington, 2018.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, ¨ You Only Look Once: Unified, Real-Time Object Detection¨, arXiv:1506.02640v5, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, ¨Deep Residual Learning for Image Recognition¨, In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, 2016.

[10] Karen Simonyan and Andrew Zisserman, ¨Very deep convolutional networks for largescale image recognition¨, arXiv:1409.1556v6, Oxford, 2015.

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov, ¨Dropout: a simple way to prevent neural networks from overfitting¨, The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, Canada, 2014.

[12] Ross Girshick, ¨ Fast R-CNN¨, arXiv:1504.08083v2, 2015.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, ¨ Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5) ¨, arXiv:1311.2524v5, California, 2014.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, ¨ Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks¨, arXiv:1506.01497v3, 2016.

[15] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan1 and Serge Belongie, ¨ Feature Pyramid Networks for Object Detection¨, In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117-2125, 2017.

[16] Xinlei Chen and Abhinav Gupta, ¨An Implementation of Faster RCNN with Study for Region Sampling¨, arXiv:1702.02138v2, Pennsylvania, 2017.

[17] Xinyi Zhou, Wei Gong, WenLong Fu and Fengtong Du, ¨Application of Deep Learning in Object Detection¨, Beijng, 2017.

[18] Yali Amit and Pedro Felzenszwalb, ¨Object Detection¨, Chicago, 2014.